THE UNIVERSITY OF ALBERTA

TRELLIS CODED MODULATION FOR HIGH BIT RATE DIGITAL SUBSCRIBER LOOPS

by

THOMAS D. BILODEAU

(C)

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

SPRING 1990

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

ISBN 0-315-60240-6

NL-339 (r. 88/04) c

Canada

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR:	THOMAS D. BILODEAU
TITLE OF THESIS:	TRELLIS CODED MODULATION FOR HIGH BIT RATE DIGITAL SUBSCRIBER LOOPS
DEGREE:	MASTER OF SCIENCE
YEAR THIS DEGREE GRANTED:	SPRING 1990

Permission is hereby granted to **THE UNIVERSITY OF ALBERTA LIBRARY** to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Author's Signature)

17124-83 Avenue
Edmonton, Alberta
T5T-0H1

Date: Jan. 10, 1990

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled TRELLIS CODED MODULATION FOR HIGH BIT RATE DIGITAL SUBSCRIBER LOOPS submitted by THOMAS D. BILODEAU in partial fulfilment of the requirements for the degree of MASTER of SCIENCE.

W. Kyle
(Supervisor)

W. Gruver

C. G. Englefield

W. Dobson

Date: 10 Jan. '90

ABSTRACT

This thesis has analyzed different transceiver structures for the proposed high-bit-rate (800 kb/s) digital subscriber loop interface which would operate on loops in the carrier serving area (CSA). The investigation has included examining the possibilities of increasing the transmission rate on these metallic loops through the use of trellis coded modulation and improved equalization algorithms. The actual analysis was accomplished through a simulation study which included models of the subscriber loops, the various transceivers and a number of noise sources.

We have compared the achievable bit rates of transceivers using 8 PAM, 16 QAM and 32 QAM signalling combined with optimal four and eight state trellis codes to the achievable bit rates of transceivers using uncoded 4 PAM, 8 AMPM and 16 QAM, respectively. These comparisons were also done with a variety of different equalizer structures. From our results, we have found that by using both a forward T/2 fractionally spaced linear equalizer and a decision feedback equalizer together, system performance can be dramatically improved over using either equalizer separately. We have also found that the uncoded 4 PAM and coded 8 PAM baseband transceivers have the best performances compared to the other uncoded and coded passband transceivers, respectively. Since the passband transceivers are considerably more complicated, the baseband transceivers are clearly the preferred transceivers to provide digital transmission over subscriber loops. However, it is only with trellis encoding that these transceivers can achieve the performance

which is necessary to ensure the ubiquitous deployment of the high-bit-rate access (800 kb/s) in the carrier serving area.

ACKNOWLEDGEMENTS

I would like to thank Dr. W. A. Krzymien for supervising this project and for many enlightening discussions. I would also like to thank the members of the examining committee, Drs. W. Dobosiewicz, C. G. Englefield and W. D. Grover for taking time to read my thesis and to suggest improvements.

I would like to acknowledge and thank Mr. A. Shen for his collaboration on the programs which modelled the digital subscriber loops and for many fruitful discussions.

I am also grateful to the staff and students of the Alberta Telecommunications Research Centre for providing an enjoyable environment to carry out this work.

For financial assistance, I am deeply indebted to the Natural Sciences and Engineering Research Council, the Alberta Telecommunications Research Centre, and the University of Alberta.

Finally, I would like to thank my family, and especially my wife Kathy, for their constant love, support and encouragement throughout this project.

- O Yeah, thanks for the typing Kath.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION....1

1.1 Integrated Services Digital Network....2

1.2 ISDN Interfaces....3

1.3 Methods of Achieving Full Duplex Transmission....8

2. IMPAIRMENTS OF DIGITAL TRANSMISSION OVER SUBSCRIBER LOOPS....11

2.1 Subscriber Loop Characteristics....11

2.1.1 Loop Configuration....11

2.1.2 Filtering....19

2.1.3 Equalization....24

2.1.3.1 Linear Equalizers....25

2.1.3.1.1 Symbol Spaced Equalizer....25

2.1.3.1.2 Fractionally Spaced Equalizer....28

2.1.3.2 Non-linear Equalizers....31

2.1.3.2.1 Decision Feedback Equalizer....31

2.1.3.2.2 Data-Aided ISI Canceller....33

2.1.3.3 Adaptive Equalization....36

2.1.3.3.1 Peak Distortion Criterion and the Zero-Forcing Adaptation Algorithm....36

2.1.3.3.2 Mean-Square-Error Criterion and the Least Mean Square Adaptation Algorithm....39

2.2 Transmission Disturbances....48

2.2.1 Echo....48

2.2.1.1 Linear Echo Canceller....50

2.2.1.2 Non-linear Echo Canceller....51

2.2.2 Impulse Noise....55

2.2.3 White Noise.....57

2.2.4 Crosstalk.....58

3. TRANSMISSION TECHNIQUES FOR THE DIGITAL SUBSCRIBER LOOP.....63

3.1 Baseband Transmission Techniques with a Non-Redundant Line Code.....63

3.2 Passband Transmission Techniques with Non-Redundant Line Coding.....67

3.3 Transmission Techniques with Expanded Signal Space Coding.....73

3.3.1 Trellis Coded Modulation.....75

3.3.2 Viterbi Algorithm Decoder.....84

3.3.3 Parallel Decision Feedback Decoding.....90

3.3.4 Performance of Trellis Codes.....93

3.3.5 Rotationally Invariant Trellis Codes.....99

4. COMPUTER SIMULATION EXPERIMENTS ON DIGITAL SUBSCRIBER LOOP TRANSMISSION.....109

4.1 Digital Subscriber Loop Model.....109

4.1.1 Calculation of the Characteristic Impedance and Propagation Constant.....109

4.1.2 Calculation of ABCD Transmission Matrices.....114

4.2 Transceiver Structure.....124

4.3 Transceiver Model.....140

4.3.1 Received Signal Model.....142

4.3.2 Echo Model.....148

4.3.3 Coloured Gaussian Noise Model.....149

4.3.4 Near-end Crosstalk Model.....150

4.3.5 Impulse Noise.....151

4.3.6 Generation and Scaling
of the Signal and Echo Impulses.....152
4.3.7 Generation and Scaling of Coloured Gaussian Noise...155
4.3.8 Generation and Scaling of NEXT.....157
4.4 Adaptive Equalizers and Viterbi Decoder.....158
4.5 Generation of System Performance Measurements.....160
4.5.1 Quantizer Signal to Noise Ratio.....160
4.5.2 Bit Error Rate Estimation.....160
5. COMPUTER SIMULATION RESULTS.....167
5.1 Results on the Farthest Reaching Loop.....170
5.1.1 Baseband Transceivers.....170
5.1.2 Passband Transceivers.....172
5.2 Results on the Worst Case Loop.....175
6. CONCLUSIONS.....178
REFERENCES.....181
APPENDIX A Derivation of Critical Frequencies.....185
APPENDIX B Tables of Primary Constants.....189
APPENDIX C Derivation of an Equivalent Series ABCD Matrix
for a Bridged Tap.....193
APPENDIX D Composite ABCD Matrices for the
Two Directions of Transmission.....195
APPENDIX E Derivation of the Transfer Functions
Involving the Subscriber Loop and the Hybrids.....198
E.1 Transfer Function of the Signal Path
through the Subscriber Loop and Hybrids.....198
E.2 Transfer Functions of the Echo Paths.....202
APPENDIX F Transfer Functions of the Simulation Experiments.....204

LIST OF TABLES

TABLE		PAGE
3.1	Bit to symbol mapping for 2B1Q line code	64
B.1	Primary constants of 19 AWG PIC at 21 degrees celsius	189
B.2	Primary constants of 22 AWG PIC at 21 degrees celsius	190
B.3	Primary constants of 24 AWG PIC at 21 degrees celsius	191
B.4	Primary constants of 26 AWG PIC at 21 degrees celsius	192

LIST OF FIGURES

FIGURE		PAGE
1.1	ISDN block diagram	5
1.2	Primary access using two 800 kb/s bidirectional lines	7
1.3	Echo cancellation method of full-duplex data transmission	9
1.4	Simplified diagram of a hybrid circuit	9
1.5	Time compression multiplexing method of full-duplex data transmission	9
2.1	Magnitude responses for 2100m of 19, 22, 24 and 26 AWG cable	12
2.2	Cable gauge distribution as a function of distance from the central office	14
2.3	A single bridged tap connected to a main cable pair	14
2.4	Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with a single 24 AWG bridged tap of varying length	16
2.5	Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with two 24 AWG bridged taps of equal but varying length	16
2.6	Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with two 450m bridged taps of varying gauge	18
2.7	Pulse responses for a 2100m 26 AWG cable with and without a bridged tap	20
2.8	(a) The raised-cosine family of pulses, for four values of α... (b) The family of spectra of the raised-cosine pulses	22
2.9	T-spaced linear transversal equalizer structure	26
2.10	Fractionally spaced linear equalizer structure	29
2.11	Decision feedback equalizer structure	32
2.12	Data-aided ISI canceller structure	35
2.13	General structure of an adaptive equalizer	47
2.14	Non-linear echo canceller with one block of RAM	52
2.15	Non-linear echo canceller with M blocks of RAM	54

2.16 Illustration of NEXT and FEXT for two digital subscriber loops.....59

3.1 (a) Signal constellation for the 8 AMPM signal.....71
(b) Signal constellation for the 16 QAM signal.....71
(c) Signal constellation for the 32 QAM signal.....71

3.2 Block diagram of a trellis encoder.....76

3.3 Set partitioning of the 8 PAM constellation.....77

3.4 (a) 8 PAM four state convolutional encoder.....79
(b) 8 PAM four state trellis diagram.....79

3.5 Set partitioning of the 8 PAM constellation showing the bits to symbols mapping.....83

3.6 Three stages of a trellis diagram.....85

3.7 An example illustrating how the Viterbi decoder functions.....88

3.8 4 and 8 PAM constellations with P_{av}=21.....95

3.9 Illustration of a minimum distance error event.....97

3.10 Signal constellation for the 4 PSK signal and the rotated constellation caused by a constant phase error.....101

3.11 Signal constellation for the 8 PSK signal.....101

3.12 (a) 32 QAM constellation showing the bits to symbols mapping and symbol groupings for an 8 state code.....105
(b) 32 QAM eight state trellis diagram.....105
(c) 32 QAM eight state non-linear convolutional encoder.....106

4.1 Lumped parameter model for a short section of transmission line.....110

4.2 General flow chart for the program INTERPOL.FOR.....113

4.3 Illustration of a two-port network with the definitions of its voltages and currents.....116

4.4 An example of a loop with three levels of bridged taps.....118

4.5 (a) Illustration of a subscriber loop's structure.....120
(b) Corresponding input file to the program.....120

4.6 General flow chart for the program DTXLINE.FOR.....121

4.7 Generic block diagram of the transceiver assumed in the simulation.....125

4.8 Uncoded 4 PAM constellation....127

4.9 Coded 8 PAM constellation with bit mapping and symbol groupings....127

4.10 (a) 8 PAM four state convolutional encoder....128
(b) 8 PAM four state trellis diagram....128

4.11 (a) 8 PAM eight state convolutional encoder....129
(b) 8 PAM eight state trellis diagram....129

4.12 Uncoded 8 QAM (AMPM) constellation....131

4.13 (a) Coded 16 QAM four state constellation....131
(b) Coded 16 QAM eight state constellation....131

4.14 (a) 16 QAM and 32 QAM four state convolutional encoder....132
(b) 16 QAM and 32 QAM four state trellis diagram....132

4.15 (a) 16 QAM and 32 QAM eight state convolutional encoder....133
(b) 16 QAM and 32 QAM eight state trellis diagram....133

4.16 Uncoded 16 QAM constellation....134

4.17 (a) Coded 32 QAM four state constellation....134
(b) Coded 32 QAM eight state constellation....134

4.18 Balancing network used within the hybrid circuit....136

4.19 Pulse responses of a 2100m 26 AWG cable with and without a pre-equalizer....138

4.20 Simulated signal processing structure for one direction of transmission....141

4.21 Confidence bands on BER when observed value is 10^{-k}....162

4.22 General flowchart of the programs simulating digital subscriber loop systems....163

5.1 (a) Configuration of farthest reaching loop....169
(b) Configuration of worst case loop....169

5.2 BER versus bit rate curves for the baseband transceivers transmitting on the loop in Figure 5.1a....169

5.3 BER versus bit rate curves for the uncoded 8 AMPM and coded 16 QAM passband transceivers transmitting on the loop in Figure 5.1a....173

5.4 BER versus bit rate curves for the uncoded 16 QAM and coded 32 QAM passband transceivers transmitting on the loop in Figure 5.1a....173

5.5 BER versus bit rate curves for the baseband transceivers transmitting on the loop in Figure 5.2a....176

5.6 BER versus bit rate curves for the uncoded 16 QAM and coded 32 QAM passband transceivers transmitting on the loop in Figure 5.2a....176

A.1 Magnitude response of a 2100m 26 AWG cable with two 450m 24 AWG bridged taps....187

C.1 Main loop with a bridged tap....194

C.2 Bridged tap as a series two-port network....194

D.1 Two-port network for the direction of transmission from the line termination to the network termination....195

D.2 Two-port network for the direction of transmission from the network termination to the line termination....197

E.1 A model for a subscriber loop and hybrids....199

CHAPTER 1

INTRODUCTION

During the last two decades telephone companies have been using growing amounts of digital transmission and digital switching equipment to transmit voice and data signals. As a result of this evolution, digital transmission has become the primary method of transmitting signals in the telecommunications network. At present, however, this integrated digital network IDN [1] does not extend all the way to the subscriber's home or office. The subscriber must send an analog voice or data signal over a twisted pair copper wire, called a subscriber loop, to a central office. Once at the central office the analog signal is digitized and then retransmitted into the IDN. This conversion is reversed once the digital signal reaches its destination central office. At this point, it is converted back to an analog signal and sent over a second subscriber loop which links the destination central office with the destination subscriber. This works well for voice communications because a voice usually starts out as an analog signal. However, for digital data communications, which now accounts for more than 10% of the network traffic in North America and which is increasing at a rate of 30% a year [2], this is a very poor architecture. A data signal from a computer or facsimile machine starts off being digital, but must be converted into an analog signal by a modem to be transmitted over a subscriber loop. Once over the loop, this analog signal must again be converted, but this time back to a digital form to be sent over the IDN. These conversions must be done at both the transmit and receive end of the IDN. Thus, two analog to digital (A/D) and two digital to analog

(D/A) conversions are necessary to send a digital data signal over the network. This is both inefficient and often unreliable. Another problem with transmitting data over an analog subscriber loop is that the capacity of the loop is very small. Consequently, the next and final step in the evolution towards an all digital network will be to make the subscriber loop digital. This extension of the digital network will provide end to end digital voice transmission and will open up opportunities for a wide variety of digital data services. This extension leads to the concept of the integrated services digital network (ISDN).

1.1 Integrated Services Digital Network

There are three main features which characterize ISDN [3]. The first of these is that ISDN promises to provide a number of existing and new communications services to its users. Some examples of these services and their required transmission rates are as follows. Low data rate telemetry services, such as home security, meter readings, and opinion polling, would need only 10 to 100 bits per second (b/s) of data. Obviously, digitized voice service would replace existing analog voice transmission and, in addition, data communications between computers and facsimile machines would no longer require modems to use the telephone network. These services would require an information rate between 10 and 100 kilobits per second (kb/s). Higher data rate applications, including high quality music and videophones, would need a transmission capacity between 100 kb/s and 1 Megabit per second (Mb/s). Finally, some very high data rate applications such as high-definition TV could be implemented using transmission rates of over 100 Mb/s.

The second feature of ISDN is that all of these services would use a standard interface. All ISDN terminal equipment would plug into a standard outlet in the same way as electrical appliances all use the same standardized 120 volt power outlet. This feature would eliminate the need for adapters and would allow the end user to connect his/her equipment to the telecommunications network without thought as to the format or protocol used.

The final feature of ISDN is end to end digital connectivity. This, in fact, is the most important characteristic of ISDN. Without it, many of the above services could not be implemented because of the demand for large data rates. Also, by having a completely digital network, the conversion of digital signals to and from analog form would be eliminated. This would result in savings to both users and telephone companies because the equipment necessary to do the data conversions would no longer be required. Thus, the successful conversion of the existing twisted-pair analog subscriber loops into digital subscriber loops will play a major role in the implementation of the ISDN.

1.2 ISDN Interfaces

The International Telegraph and Telephone Consultative Committee (CCITT) is an international organization which defines technical issues for ISDN. It has developed a set of standards, part of which defines two major access interfaces. The first, which will be the most common, is the basic access interface. It will provide the subscriber with two 64 kb/s full duplex (simultaneous bidirectional transmission) B channels and one 16 kb/s D channel. As a result, the data rate required for this 2B+D transmission is 144 kb/s; this will be provided on the currently

existing subscriber loops. Each B channel can be used as a pulse code modulated (PCM) voice channel or as a 64 kb/s data channel. The 16 kb/s D channel will be used for network control and for low speed data transmission [1].

The second ISDN access interface standard, which will be used by larger customers, is the primary rate access interface. This interface will have one of two different structures depending upon the country it is installed in. North America and Japan will use a 1.544 Mb/s system which will carry 23 B and 1 D channels. Most other countries will use a 2.048 Mb/s system consisting of 30 B and 1 D channels. These transmission rates were chosen to be compatible with existing digital trunk formats used in various countries [4]. One should note that a primary rate access interface uses a 64 kb/s D channel instead of the 16 kb/s D channel used in basic access interface. Full duplex primary rate access will not be provided on all of the existing twisted pair subscriber loops. However, there is a subset of loops which have the necessary channel capacities to carry a primary rate access signal, as long as transceivers with sufficient complexity can be built [5].

Four reference points or physical interfaces have also been defined by the CCITT. They are the R, S, T and U interfaces. The S and U interfaces are the two most important physical interfaces and are shown in Figure 1.1. The S interface is located at the customer's home or office and uses two separate wire pairs for the two directions of transmission. Data can be sent from terminal equipment (TE) to the network termination (NT) in either a point to point fashion or by means of a multipoint connection bus as again shown in Figure 1.1 [6]. The U interface is the physical point of connection between the twisted pair

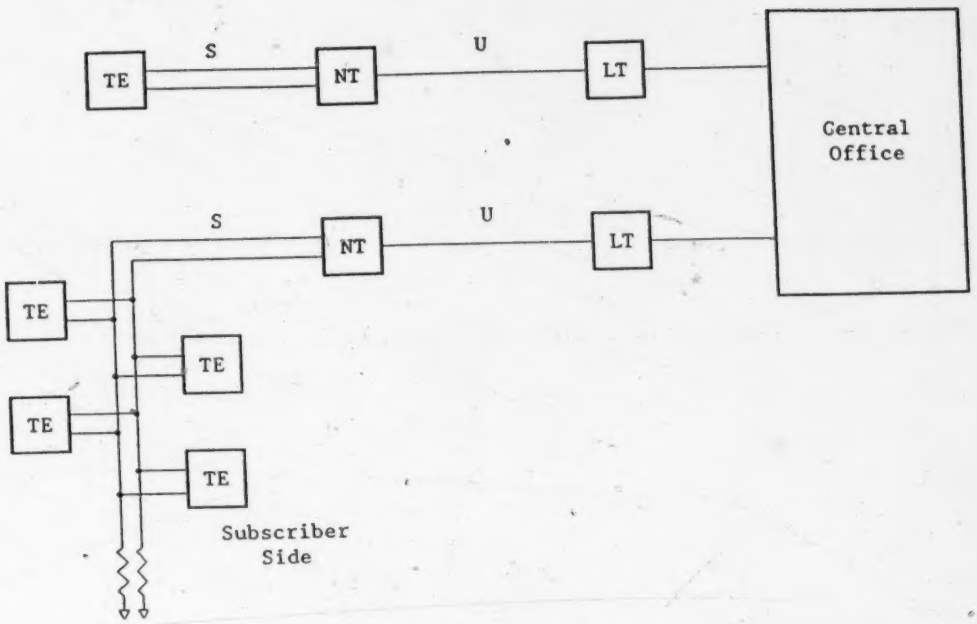


Figure 1.1 ISDN block diagram

and the customer's NT. It provides the basic access full duplex digital signal which travels on the subscriber loop between the NT and the central office's line termination (LT). As stated previously, this basic access (2B+D) signal requires a data rate of 144 kb/s. However, the transmission rate of the digital subscriber loop must be higher to provide for framing and maintenance functions. Thus, the American National Standards Institute's (ANSI) standard U-interface transmission rate is 160 kb/s where 16 kb/s of transmission overhead is provided. For ISDN to be economically feasible, 99% of the subscriber loops must be able to support at least this bit rate at bit error rates (BERs) less than 10^{-7}. This equates to operation over loops which are up to 5.5 kilometres (km) long and have not been conditioned [7].

More recently, a high-bit-rate digital subscriber access has been proposed in a document by J.W. Lechleider [8]. He argues that with advancing very large scale integration (VLSI) technology, highly complex single chip transceivers will be possible in the near future. Consequently, he suggests that an interface with a bit rate on the order of 800 kb/s, and operating on a subset of the existing subscriber loops be studied. This subset of loops, called the Carrier Serving Area (CSA), surrounds a central office or a remote terminal. The loops within this CSA are restricted to a total length of less than 3650 m, including the length of bridged taps (see section 2.1.1), and do not contain any 26 AWG cable . If there is 26 AWG cable in the loop, then its total length including bridged taps must be less than 2750 m . In addition to these two constraints, the total length of all the bridged taps off a main cable pair must be less than 760 m with no single bridged tap longer than 610 m .

The figure of 800 kb/s was arrived at after considering the improvements that might be achieved by using more sophisticated signal processing in the digital subscriber loop transceivers. This signal processing could include either, or both, better equalization and the use of coding to improve performance. Another possible argument for the high-bit-rate lines, discussed in Lechleider's proposal, is to use two of these 800 kb/s bidirectional loops in parallel to form a primary access channel as shown in Figure 1.2. This configuration would have the feature that any two unconditioned loops in the CSA could be used to provide primary rate access. A third argument, for a digital subscriber access interface which would transmit at a data rate significantly greater than the basic access interface data rate, is as follows. It has been shown [5],[9],[10] that the channel capacity of the subscriber lines varies considerably with length and for short loops significantly exceeds 1 Mb/s. By deploying the basic access interface, at a rate of 160 kb/s, there would be a dramatic under-utilization of the transmission capacity available. This situation would be substantially improved if a transmission rate of around 800 kb/s was used to transmit data on digital subscriber loops.

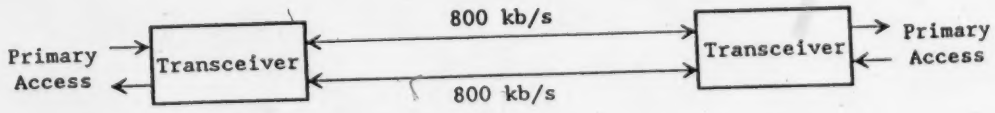
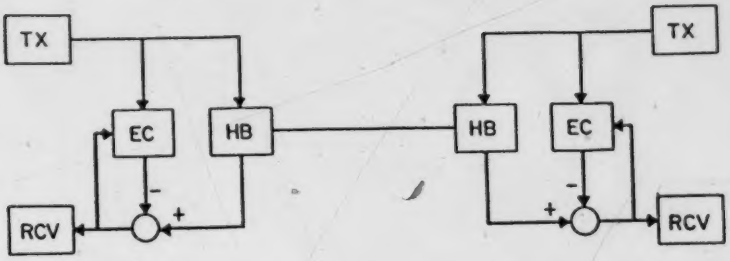


Figure 1.2 Primary access using two 800 kb/s bidirectional lines

1.3 Methods of Achieving Full Duplex Transmission

There are two techniques which can be used to realize full duplex digital transmission on a single wire pair. These methods are echo cancellation (EC) and time compression multiplexing (TCM). In a transmission system using an echo canceller, shown in Figure 1.3, data flows in both directions simultaneously. As a result, the line rate will be equal to the data rate, which, for basic access, is equal to 160 kb/s. The EC method uses a hybrid circuit, Figure 1.4, to separate the two directions of transmission by performing a two wire to four wire conversion. However, due to the non-ideal impedance matching of the hybrid's balancing network to the line, some leakage of the near-end transmitted signal into the near-end received signal occurs. This leaked signal is called an echo. An adaptive echo canceller, which can be quite complex, is used to cancel this unwanted signal.

TCM, the burst method, or the ping-pong method, shown in Figure 1.5, achieves full duplex transmission by transmitting a burst of information, at an accelerated rate, in one direction at a time. This, in principle, would require that the actual transmission rate within bursts be twice the data rate. However, guard times are also inserted between the two directions of transmission so that near-end transients do not affect the reception of far-end pulses. Consequently, for a basic access data rate of 160 kb/s the actual transmission rate required within each burst is approximately 2.3 times higher (368 kb/s) [6]. This method of converting from two wires to four wires uses relatively simple hardware compared to the EC method. Only a transmit/receive switch is required at the loop interfaces to switch from the transmit wire pair to the receive wire pair and back at proper instants. Thus,



TX = TRANSMITTER
RCV = RECEIVER
EC = ECHO CANCELLER
HB = HYBRID

Figure 1.3 Echo cancellation method of full-duplex data transmission

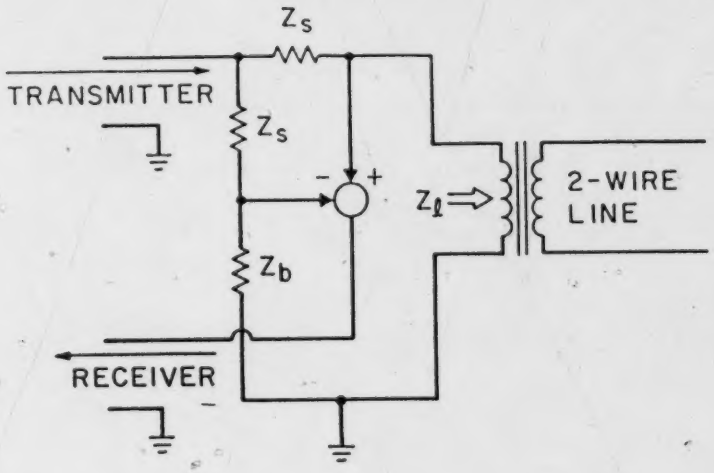


Figure 1.4 Simplified diagram of a hybrid circuit

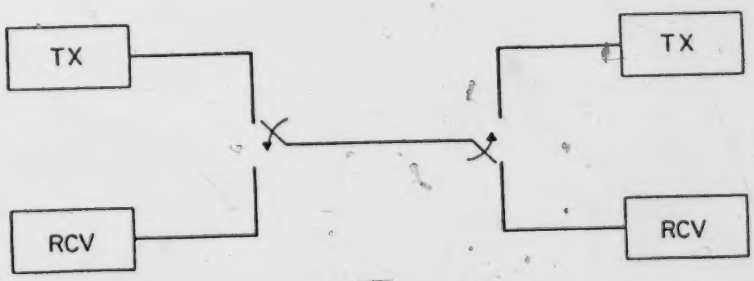


Figure 1.5 Time compression multiplexing method of full-duplex data transmission

the TCM method causes an increase in the transmission rate, but does not require a complex echo canceller.

Due to the different noise environments and loop plant characteristics, the EC method has become the preferred method of full duplex transmission in North America and Europe, while the TCM method is used in Japan [11]. The work contained in this thesis uses the EC method since it is the preferred method in North America.

Chapter 2

Impairments of Digital Transmission over Subscriber Loops

In order to design and analyze a transceiver for a digital subscriber loop system, it is necessary to understand the transmission properties of the subscriber loop and the disturbances present in its environment. Once this knowledge is in hand, one can then find ways of minimizing or eliminating the non-idealities they cause. Thus, the main sources of impairments in a digital subscriber loop and several ways of mitigating their affects will be discussed.

2.1 Subscriber Loop Characteristics

2.1.1 Loop Configuration

A number of surveys have been performed over the last two decades to determine the physical and electrical characteristics of subscriber lines. One observation made from these surveys is that no two loops are the same. Loops are typically made up of sections which vary in length and in gauge. Some worst case loops have been found with up to 30 gauge changes. The most common gauges of cable used within the subscriber loop plant are 19, 22, 24 and 26 American Wire Gauge (AWG) polyethylene insulated cable (PIC). The 26 AWG cable has the smallest diameter (0.404 mm) and thus the highest per unit length resistance. The 24, 22 and 19 AWG cables have increasingly large diameters (0.511 mm, 0.645 mm and 0.912 mm, respectively) and thus decreasingly small per unit length resistances. This results in 26 AWG cable attenuating signals more than an equal length of a larger diameter cable. This can be seen in Figure 2.1, where there is a 16 dB difference in attenuation between the 19 and

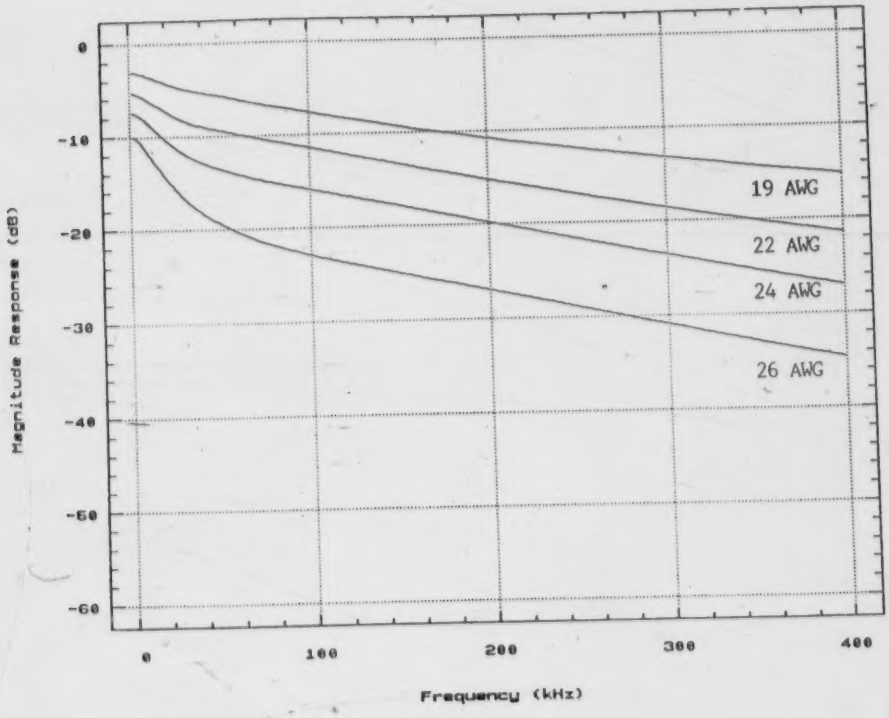


Figure 2.1 Magnitude responses for 2100m of 19, 22, 24 and 26 AWG cable

26 AWG cables (both 2100 m long) at 200 kHz. From Figure 2.1, we also see what the frequency response of a uniform gauge section of wire looks like. The attenuation curve starts out with quite a steep slope for low frequencies, but as the frequency increases, the attenuation in dB becomes approximately proportional to the square-root of frequency. This results in a subscriber loop acting as a low pass filter with increasingly large passband attenuation.

A second important characteristic from the subscriber loop surveys is the wire gauge distribution as a function of distance shown in Figure 2.2 [12]. It can be seen that as the distance away from the central office increases, the dominance of the 26 AWG cable is slowly replaced by the 24 and then the 22 AWG cable. The 19 AWG cable only plays an important role in very long subscriber loops. The reason for this gauge distribution is that telephone companies have tried to keep the total d.c. resistance of any particular subscriber loop below 1300 ohms. Thus, shorter loops can use the 26 AWG cable with its high per unit-length resistance while longer loops must be constructed out of sections of 24, 22 and 19 AWG cable, which have the smaller per unit length resistances. From the 1981 Bell Canada survey an average loop length of 3.102 km and average resistance of 667 ohms was found.

Another physical characteristic of the subscriber loop plant is the existence of open-circuited cable pairs which are connected in parallel to the main cable pair as shown in Figure 2.3. These open-circuited cables are called bridged taps. It is common for a subscriber loop to have two or more bridged taps of unknown length attached to it. These taps are used to reach several geographically dispersed locations so that each main pair could be used by anyone of several possible

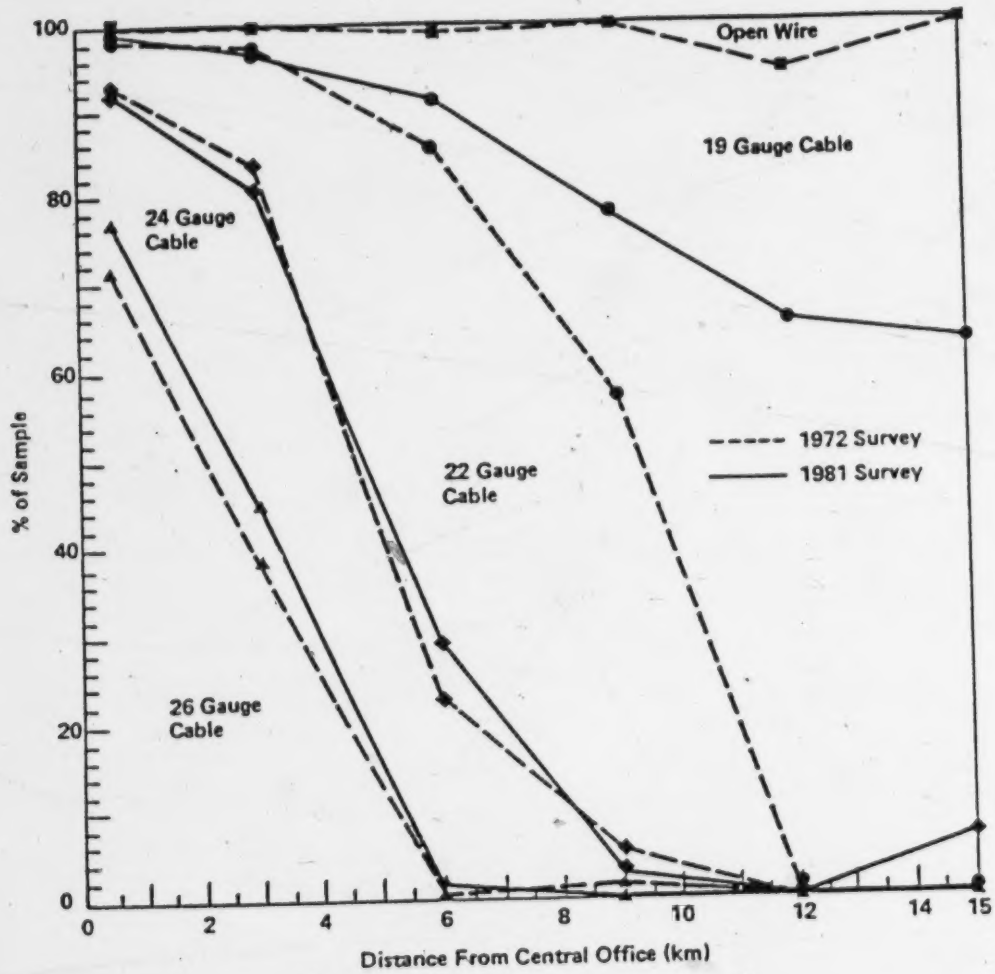


Figure 2.2 Cable gauge distribution as a function of distance from the central office

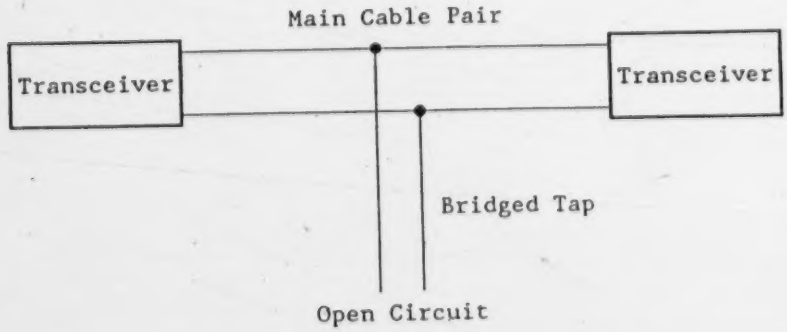


Figure 2.3 A single bridged tap connected to a main cable pair

subscribers. In this way, if one subscriber no longer needs his/her line, another subscriber can use it. Thus, the telephone company ensures that as many of its installed pairs are being used as possible, keeping its fill ratio close to unity. (Fill ratio is the ratio of the number of working pairs to the total number of installed pairs [13].)

Bridged taps cause a number of problems on digital subscriber loops. First, additional attenuation at every frequency occurs because the taps load the cable. This can be seen in Figure 2.4, where the frequency responses of a main cable without a bridged tap and a main cable with a single bridged tap of varying length are shown. Without regard to the length of the bridged tap, its presence on the main cable pair causes at least some if not a significant amount of additional attenuation at every frequency.

A second problem, caused by the bridged taps, is frequency-selective attenuation. If we look again at Figure 2.4, dips in the frequency response occur at one or several frequencies depending on the length of the tap. These dips come as a result of the length of a tap approaching a quarter wavelength ($\lambda/4$) of some frequency and the main cable then seeing the tap as a short circuit. In fact, if the length of a tap is equal to $\lambda_1/4$ and $3\lambda_2/4$ and $5\lambda_3/4$ etc., the tap will cause multiple dips in the frequency response as shown in Appendix A. From Figure 2.4, we see that the shorter bridged taps result in dips at higher frequencies (150 m tap causes a dip at 310 kHz) while the longer taps cause dips at lower frequencies (750 m tap has its first dip at 62 kHz).

When there are several bridged taps attached to a main line, their effects are approximately additive. This can be seen in Figure 2.5,

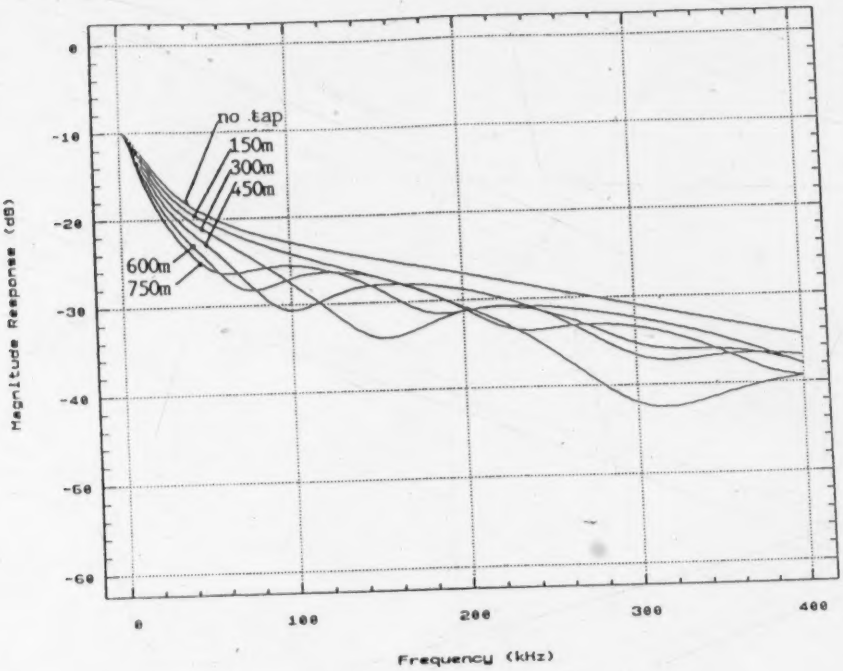


Figure 2.4 Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with a single 24 AWG bridged tap of varying length

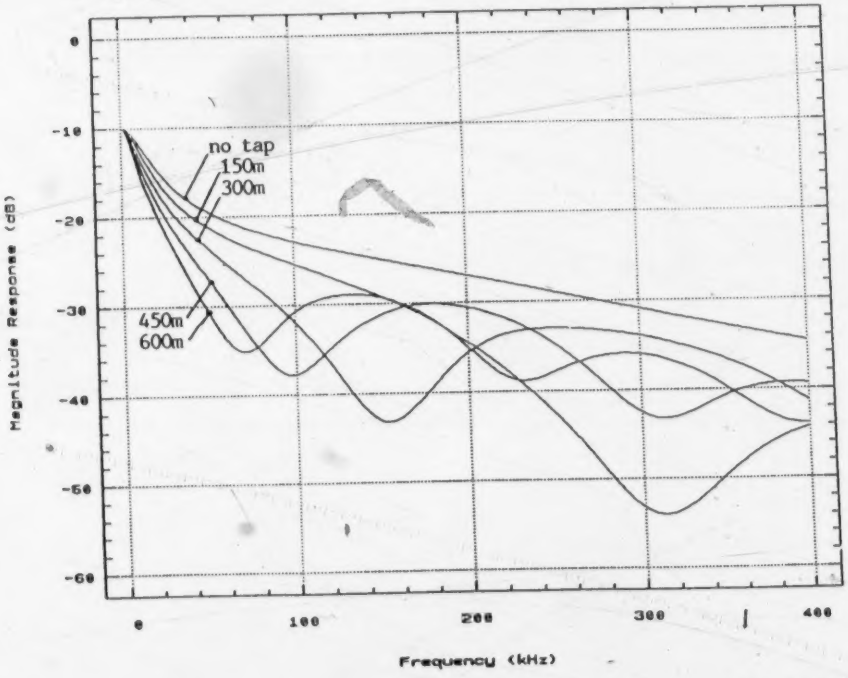


Figure 2.5 Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with two 24 AWG bridged taps of equal but varying length

where two bridged taps of the same length are placed on a main cable pair. The dips caused by the two bridged taps are significantly deeper than the corresponding dips caused by a single bridged tap. It should be noted that two taps, having the same length, attenuate the same frequencies and as a result cause a "maximum" dip at their critical ($\lambda/4$, $3\lambda/4$, etc) frequencies.

Another observation is that the gauge of the bridged tap also plays an important role in the amount of attenuation it causes. As shown in Figure 2.6, the smaller the gauge (larger the wire diameter) of the bridged tap, the more attenuation it causes. This is reasonable since the smaller the gauge, the smaller the per unit length resistance and the closer the bridged tap is to a lossless quarter wavelength line. A lossless quarter wavelength line would not just attenuate the signal, but would eliminate the signal at that frequency.

A third problem, caused by bridged taps, is that they cause reflections towards both the transmitter and the receiver. When a pulse comes to a bridged tap, part of its energy continues down the main line and part of its energy goes down the tap. This split energy on the tap will continue to propagate until it reaches the open circuit. Here it will be reflected back towards the junction between the bridged tap and the main line from which it came (unless the bridged tap is a quarter wavelength long). Once this energy, in the form of a pulse, reaches the junction it again splits this time part of it heading towards the transmitter and part of it towards the receiver. The pulse which heads towards the transmitter will become part of the echo noise which must be removed by an echo canceiler. This is discussed in section 2.2.1. The pulse which heads towards the receiver acts as a "ghost pulse" which is

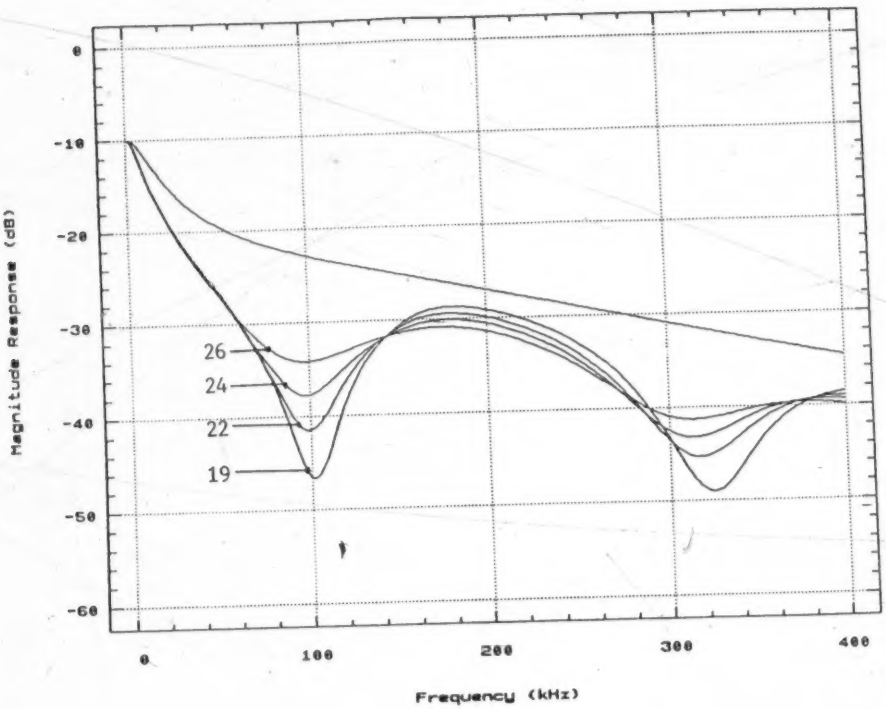


Figure 2.6 Magnitude responses for a 2100m 26 AWG cable without a bridged tap and with two 450m bridged taps of varying gauge

a delayed and attenuated version of the main received pulse. This is shown in Figure 2.7. The first pulse response is for a main cable without a bridged tap. It simply contains a main pulse followed by a long tail. The second pulse response is for a subscriber loop with a bridged tap. It shows an attenuated main pulse which is followed two and one half periods later by a "ghost pulse". This "ghost pulse" results in post-cursor distortion and must be eliminated.

By examining the previous figures we can come to a number of conclusions about the characteristics of digital subscriber loops. First, the loops cause an overall attenuation and a frequency selective attenuation which result in received pulses with smaller amplitudes. Second, long post cursor tails, as shown in Figure 2.7, are caused by loops having a non-linear phase characteristics at low frequencies. Finally, because the channels are bandlimited, pulses are dispersed or spread out in time. All these characteristics contribute to the observation that neighbouring pulses overlap and interfere with one another. This effect is known as intersymbol interference (ISI). When uncontrolled ISI is present in a system, a serious degradation in performance occurs. This loss in performance can come as either an increased BER or as a reduction in the achievable data rate for a given BER. Consequently, methods of eliminating or at least reducing ISI must be used in digital subscriber loop transceivers.

2.1.2 Filtering

The two best ways of removing ISI between pulses are proper filter design and channel equalization. The filters are designed in advance to satisfy the first Nyquist criterion. This criterion is a condition,

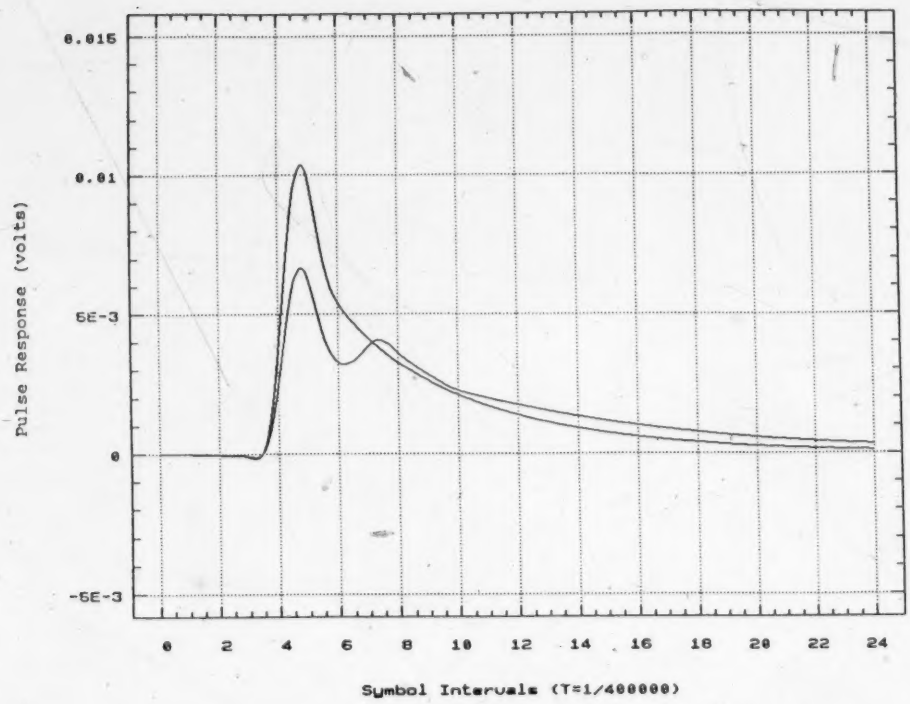


Figure 2.7 Pulse responses for a 2100m 26 AWG cable with and without a bridged tap

placed on the received pulses of a system, which may be stated as: a pulse-shape p(t), which has the Fourier transform P(f) must satisfy the frequency domain condition [14]

$$\sum_{m=-\infty}^{\infty} P(f+m/T)=T, \qquad |f|\leq 1/2T \tag{2.1}$$

and leads to the time domain condition that

$$p(nT)=\begin{cases} 1 & n=0 \\ 0 & n\neq 0 \end{cases} \tag{2.2}$$

where T is the symbol rate. If we represent the received data stream by

$$r(t)=\sum_{m=-\infty}^{\infty} a_m p(t-mT) \tag{2.3}$$

where a_m are the real or complex valued transmitted symbols and sampling occurs at multiples of T seconds, then to satisfy the first Nyquist criterion, the leading and lagging tails of a pulse have to have null values at the peak or cursor positions of its neighbours. Pulses which satisfy this criterion are called Nyquist pulses.

One family of Nyquist pulses are the raised-cosine pulses, given by

$$p(t)=\left[\frac{\sin(\pi t/T)}{\pi t/T}\right]\left[\frac{\cos(\alpha\pi t/T)}{1-(2\alpha t/T)^2}\right] \tag{2.4}$$

where α is called the roll-off factor and is a measure of excess bandwidth. The pulse shapes, for four values of α, are shown in Figure 2.8a. Here, we can see that at t=0 the pulses have an amplitude of unity, but at all other multiples of T there are zero crossings. The family of spectra of the raised-cosine pulses is shown in Figure 2.8b. Their analytical description is given by

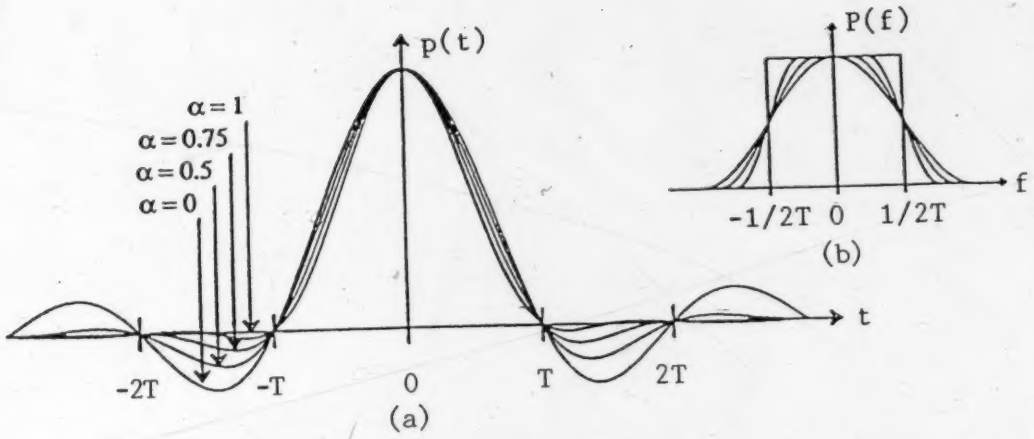


Figure 2.8 (a) The raised-cosine family of pulses, for four values of α
(b) The family of spectra of the raised-cosine pulses

$$P(f)=\begin{cases} T & 0\le|f|\le(1-\alpha)/2T \\ T/2\ [1-\sin(\pi T(f-1/2T)/\alpha)] & (1-\alpha)/2T\le|f|\le(1+\alpha)/2T \\ 0 & |f|>(1+\alpha)/2T \end{cases} \qquad (2.5)$$

Here one can see where the terms excess-bandwidth and roll-off factor come from. When α=0, the resulting pulse shape, in the time domain, is the minimum bandwidth sinc pulse which requires no excess-bandwidth for transmission. When α is changed from 0 to 1, the excess-bandwidth increases from 0% to 100% and the energy in the frequency domain rolls off more gently. At α=1, twice the minimum bandwidth is required to send the pulse.

There are a number of trade-offs when choosing an α for a system. A smaller excess bandwidth filter will result in less noise entering the receiver. However, the smaller excess-bandwidth pulse will have a longer tail with larger amplitudes. This would make the performance of the system much more sensitive to the sampling phase at the receiver. For instance, any timing jitter which would cause the sampling point to move from its ideal location would result in a significant amount of ISI in the sample. Another advantage to the shorter pulses, is that they are easier to implement with a finite impulse response (FIR) filter because they can be truncated (at some multiple of T) sooner than the longer pulses.

It must now be emphasized that the first Nyquist criterion and thus the shape of the pulses is a condition that must be met at the receiver for no ISI to occur. Because digital subscriber loops cause amplitude and phase distortion, Nyquist pulses sent by the transmitter will not satisfy Nyquist's first criterion at the receiver. The

unpredictable distortion and resulting ISI introduced by the channel will have to be compensated for by an equalizer.

2.1.3 Equalization

An equalizer is simply a receiver filter which flattens (equalizes) the amplitude response of a channel and makes the phase response a linear function of frequency over a frequency band of interest. In other words, the equalizer tries to undo any frequency dependent amplitude and phase distortion caused by the channel, so that the cascade of the channel and the equalizer results in an overall channel response which does not cause any degradation to the transmitted signal. Consequently, if the equalizer is performing well, Nyquist pulses sent by the transmitter will still be Nyquist pulses after the equalizer and no ISI will be present at the sampling point. However, often the price paid for eliminating ISI is an increase in the noise power at the decision device. This will be discussed when both linear and non-linear equalizer structures are presented in the following subsections.

Another problem that must be considered when choosing an equalizer is the uniqueness of each digital subscriber loop. This uniqueness results in a wide variety of frequency responses, all of which the equalizer will have to be able to compensate for. The solution to this problem is to make the equalizer adaptive. In this way the equalizer can be automatically custom fit to each loop. The least-mean-square (LMS) adaptation algorithm was used to adapt the equalizers in this study. Consequently, it will also be presented in the following subsections.

2.1.3.1 Linear Equalizers

2.1.3.1.1 Symbol Spaced Equalizer

Once the received signal has passed through a matched filter, an equalizer is necessary to remove ISI caused by the channel. Among all the structures used for equalization, the simplest, in concept, is the symbol or T-spaced linear transversal equalizer shown in Figure 2.9. As can be seen in the figure, the output of the matched filter is sampled at the baud rate and then these samples are shifted through the T-spaced equalizer. Within the equalizer, current and past values of the sampled received signal are linearly weighted by the equalizer coefficients (tap gains), c_n, and then summed to produce an output. The T-spaced equalizer output samples can be expressed as

$$y(k)=\sum_{n=-N1}^{N2} c_n x(kT-nT+t_o) \qquad (2.6)$$

where N1+N2+1 is the number of equalizer coefficients and t_o denotes the sampling phase. The equalizer will introduce a delay of N1 baud intervals between the received input samples and their corresponding equalized output samples. This delay permits precursor ISI to be reduced or eliminated by the equalizer.

In practice, there are several problems with a symbol spaced linear equalizer. The first problem is that in order for the receiver to be optimum, a matched filter is needed to precede the linear equalizer. To do this implies that there is a good knowledge of the channel. However, when dealing with digital subscriber loops, no two loop configurations are the same. Thus, it would be impossible and uneconomical to have a unique matched filter for every subscriber loop.

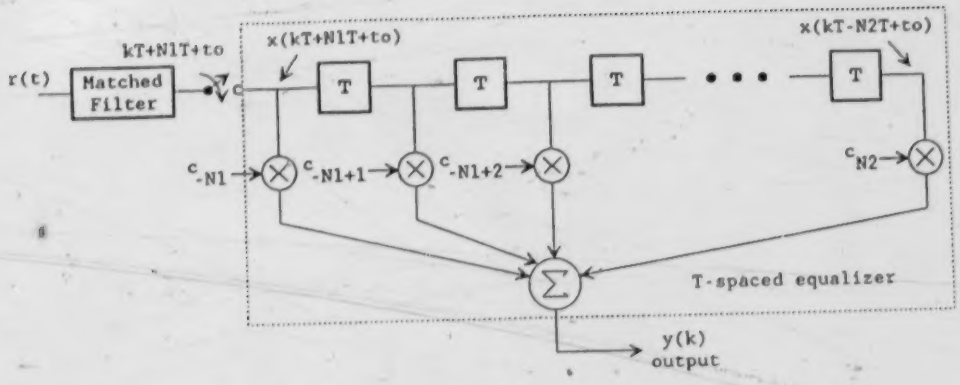


Figure 2.9 T-spaced linear transversal equalizer structure

A second problem with the T-spaced equalizer comes as a result of sampling the output of the matched filter at the baud rate. If, for simplicity, we assume that the channel has an excess bandwidth of 100%, then the sampling theorem tells us that we should sample the signal at at least twice the baud rate if no aliasing is to occur. Consequently, by operating the sampler at the baud rate a spectral overlap or aliasing of frequency components occurs. When the phase of these overlapping components match they add constructively and the equalizer must attenuate these constructive frequencies. However, when the phases are 180° apart the overlapping frequency components add destructively and produce nulls in the frequency spectrum. The equalizer compensates for this by introducing gain at these attenuated frequencies and thus we end up with what we desired, a constant amplitude response. The down side of this approach is that any noise present at the frequencies which must be amplified will also experience this same amplification. This will result in an increase in the noise power at the decision device and will cause the probability of error to be larger. This increase in noise power is called noise enhancement.

A third problem with the symbol spaced linear equalizer is that its performance is very sensitive to the sampling phase. This problem is related to the previous problem of aliasing. When the sampling phase experiences variations, it causes the phases of the overlapping components to change. This can result in changes in the depths of the nulls which the equalizer may not be able to manipulate into a flat spectrum at all, or at least without significant noise enhancement. One solution to these problems is to use a fractionally spaced linear equalizer.

2.1.3.1.2 Fractionally Spaced Equalizer

A fractionally spaced equalizer (FSE) is shown in Figure 2.10. Its structure is very similar to the symbol spaced equalizer shown in Figure 2.9 with the exception of a few important changes. First, the matched filter has been replaced by a low pass filter, which simply ensures the signal is band limited and the out of band noise is removed. The second change is that the sampler operates at a rate which is at least twice the bandwidth occupied by the signal. This is done to satisfy the sampling theorem. In practice, the sampling rate is often chosen to be twice the baud rate. The third change, which is within the equalizer itself, comes as a result of the increased sampling rate. The delay line taps are now spaced at an interval τ which is less than the symbol interval T. The interval τ will be equal to the reciprocal of the sampling rate. Thus, when sampling at twice the baud rate, τ=T/2. In a digital implementation τ must equal KT/M, where K and M are integers and M > K. With K=1 and M=2, we get τ=T/2 and our fractionally spaced equalizer can be called a T/2-spaced equalizer. Finally, the output of the FSE is sampled once per a symbol interval (sampled at the baud rate) to give an output described by

$$y(k)=\sum_{n=-N1}^{N2} c_n x(kT-nKT/M+t_o) \qquad (2.7)$$

The delay introduced by this equalizer is K*N1/M baud intervals.

Most of the problems with the symbol spaced equalizer can be solved by virtue of the FSE's increased sampling rate. By sampling the input signal fast enough to prevent aliasing, the FSE is able to adjust the signal's spectrum in amplitude and in phase before symbol rate sampling and the spectral overlap it causes. Thus, frequency

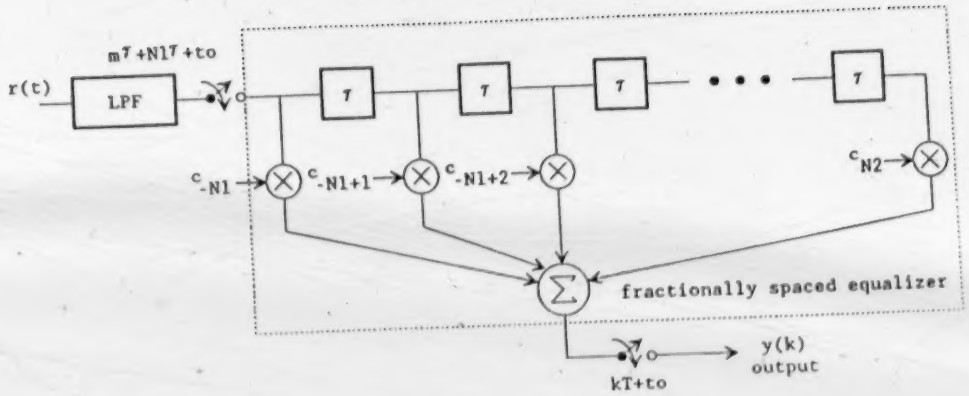


Figure 2.10 Fractionally spaced linear equalizer structure

components, which would have added destructively and caused a spectral null, can now have their phases adjusted before aliasing and in a manner which will produce the best possible overlapped combination of these components. This significantly improves the problem of noise enhancement found with the T-spaced equalizer. In addition to this, sampling phase variations can also be dealt with before aliasing and thus the FSE is far less sensitive to the sampling phase. Finally, the matched filter required by the T-spaced equalizer can be thrown away. The FSE is able to duplicate the functionality of a matched filter because of its sampling rate and leaving the spectral aliasing until the end. It should be noted that a T/2-spaced equalizer with the same number of taps as a symbol spaced equalizer will span half the time interval. Nevertheless, it has been shown that when both types of equalizers have the same number of taps the FSE performs at least as well as the symbol spaced equalizer on simple channels and performs better on channels with severe distortion [15].

Although the FSE has been shown to have a number of advantages over the symbol spaced equalizer, its performance is still not optimum when amplitude distortion causes the ISI. Often, severely distorted channels will still have nulls which will require some frequency amplification and thus noise enhancement when equalized. The performance of these linear equalizers can be improved upon by introducing some non-linearity into the equalizers.

2.1.3.2 Non-linear Equalizers

2.1.3.2.1 Decision Feedback Equalizer

The decision feedback equalizer (DFE) is the simplest of non-linear equalizers and is shown in Figure 2.11. As can be seen, the DFE consists of two transversal filters. The first filter is called the forward filter and can be either a symbol spaced or a fractionally spaced equalizer. The second filter is in the feedback path and is thus called the feedback filter. It is always a symbol spaced filter. The DFE works as follows. Samples of the received signal are linearly weighted by the forward equalizer as was shown in the previous section on linear equalizers. However, the forward equalizer is designed to eliminate the precursor ISI only, that is , to eliminate the ISI before the main lobe. At the same time, the feedback filter uses previous symbol decisions to generate a replica of the postcursor ISI caused by the tails of previously received symbols. This replica ISI is then subtracted from the output of the forward equalizer to produce the final decision feedback equalized output. The DFE output is given by

$$y(k)= \sum_{n=-N1}^{N2} c_n x(kT-nKT/M+t_o) - \sum_{m=1}^{N3} b_m d_{k-m} \qquad (2.8)$$

where N3 is the number of taps in the feedback filter, b_m are its tap values and d_{k-m} are previously made symbol decisions.

The performance advantage of the DFE comes as a result of using subtraction, instead of multiplication to deal with amplitude distortion. Rather than multiplying attenuated frequency components by a gain to remove ISI, the feedback filter of the DFE generates a replica of the postcursor ISI and subtracts it from the received signal to exactly cancel (assuming past decisions are correct) the actual

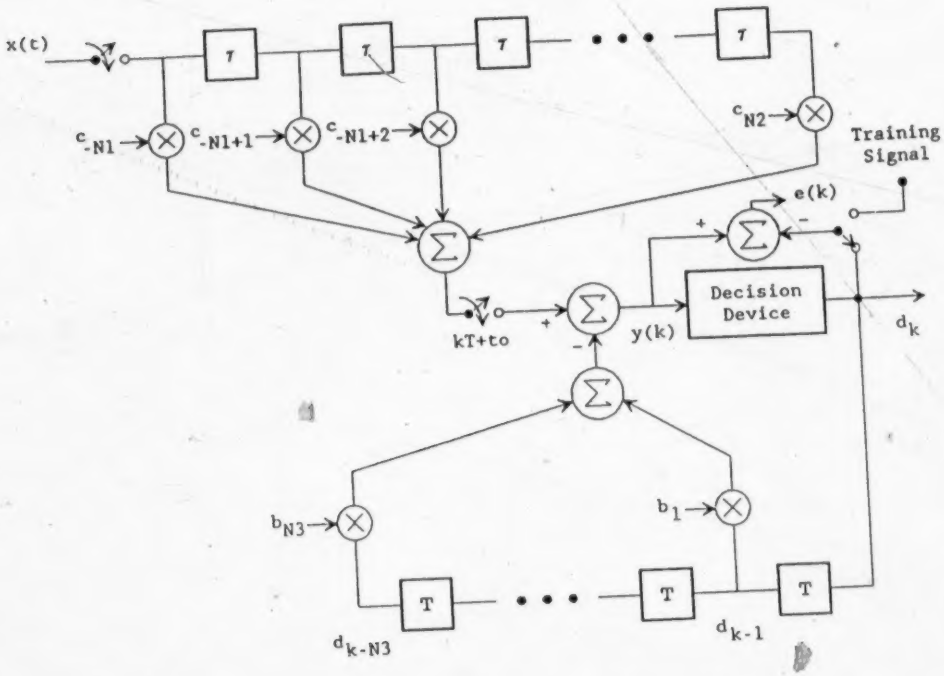


Figure 2.11 Decision feedback equalizer structure

postcursor ISI. By subtracting, instead of multiplying, the noise power received with the signal is not amplified and thus there is no noise enhancement from this part of the equalization. However, some noise enhancement is caused by the forward equalizer which uses linear equalization techniques to eliminate the precursor ISI.

One problem with the DFE comes to light when a decision error is made by the decision device. This incorrect symbol decision will be sent through the feedback filter where it will be used to create replicas of postcursor ISI. Obviously, these ISI replicas will not be correct and could cause future symbol decisions to be in error. This effect is known as error propagation. Fortunately, error propagation in a DFE is not catastrophic. Typically, it only causes a slight degradation in performance which is much smaller than the benefits that come from the reduced noise enhancement.

2.1.3.2.2 Data-Aided ISI Canceller

From the above discussion on DFEs, it was explained that subtracting or cancelling postcursor ISI from a signal had significant advantages in terms of noise enhancement. It was also explained that a forward equalizer was necessary to eliminate precursor ISI but a price was paid in terms of an increase in noise power. At this point, it would seem natural to extend the concept of ISI cancellation to include future symbols, so that both precursor and postcursor ISI could be cancelled in a similar fashion. That is, if both the past and future symbols that contribute to ISI at a sample point were known, then exact replicas of all the ISI could be generated and subtracted to leave a perfectly equalized signal with no noise enhancement. Sounds good, the

problem, however, is getting the future symbols. One solution to this problem is to have two separate equalizers working in the receiver as shown in Figure 2.12. The two equalizers get the same sequence of received samples, except the second equalizer's input sequence is delayed with respect to the first. This is done so that the first equalizer can make tentative symbol decisions, in advance, which can be used by the second equalizer as future symbol values. The first equalizer could be any one of the linear equalizers or the DFE already discussed. The second equalizer is the data-aided ISI canceller [16]. The ISI canceller's structure is identical to that of a DFE except for an additional T-spaced filter which generates a replica of the precursor ISI from the tentative decisions made by the first equalizer. The canceller's output may be given by

$$y(k)=\sum_{n=-N1}^{N2} c_n x(kT-nKT/M+t_o) - \sum_{m=1}^{N3} b_m d_{k-m} - \sum_{p=1}^{N4} g_p d_{k+p} \qquad (2.9)$$

where N4 is the number of taps in the precursor filter, g_p are its tap values and d_{k+p} are the tentative decisions made by the first equalizer.

The performance of the ISI canceller theoretically approaches that of the zero-ISI matched filter. That is, all the ISI is gone and there is no noise enhancement. However, one problem with the ISI canceller is the increase in its final probability of error that comes as a result of tentative decision errors made by the first equalizer. The severity of this problem has not been resolved, at least in published literature. In addition to this, any performance gained by using this structure would have to be weighed against the additional hardware cost of having two equalizers, instead of one, doing this part of the signal processing.

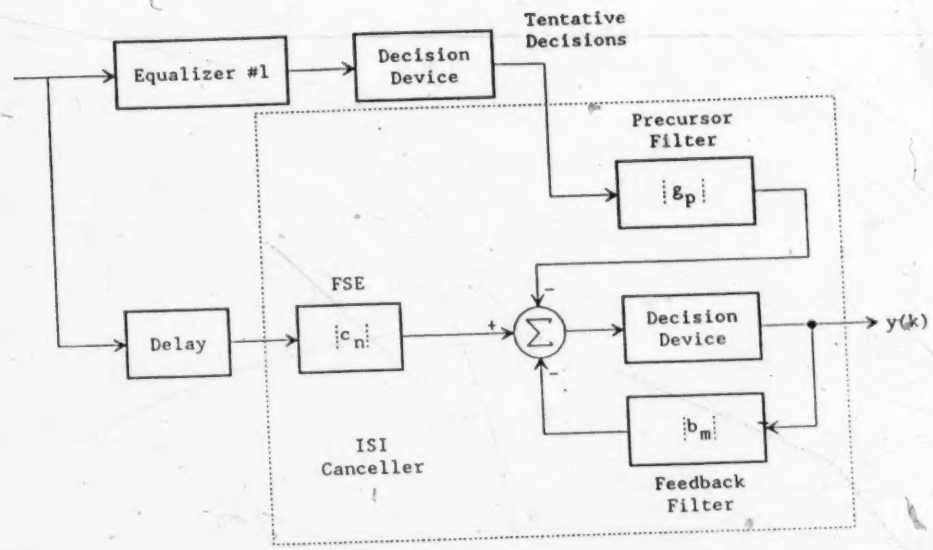


Figure 2.12 Data-aided ISI canceller structure

2.1.3.3 Adaptive Equalization

As was mentioned in the beginning of this section on equalization, there can be large variations in the frequency responses of digital subscriber loops. In order to effectively eliminate or reduce ISI on all these different channels, we must use some form of adaptive equalization. An adaptive equalizer can take the structure of any of the previously described equalizers. However, its main feature is that its coefficients can change or adapt to uniquely equalize each channel. An adaptation algorithm is used to change the coefficients so that the equalizer can minimize some pre-determined measure of error. Two criteria which minimize two different measures of error are the peak distortion criterion and the mean-square-error criterion.

2.1.3.3.1 Peak Distortion Criterion and the Zero-Forcing Adaptation Algorithm

We may begin this discussion by considering the operation of an infinite length symbol spaced linear equalizer. Its input is the sequence of samples x(k) given by

$$x(k)=\sum_{j=-\infty}^{\infty} h_j a(k-j)+\eta(k) \qquad (2.10)$$

where a(k) are the transmitted data symbols, $\eta(k)$ are samples of an additive noise and h_j are samples of the impulse response of the equivalent discrete-time channel. This equivalent channel is made up of the actual channel, a matched filter at the receiver and the baud rate sampler. The output of the equalizer y(k) is given by the convolution

$$y(k)=\sum_{n=-\infty}^{\infty} c_n x(k-n) \qquad (2.11)$$

where c_n are the equalizer's coefficients. This expression is very similar to Eqn.(2.6) except the sampling period and timing phase are not shown. At this point, it will be informative to define a new composite equivalent discrete-time channel which includes both the response of the previous equivalent channel and also the response of the equalizer. The new equivalent discrete-time channel is given by

$$q_j = \sum_{n=-\infty}^{\infty} c_n h_{j-n} \tag{2.12}$$

where q_j is simply the convolution of the sequences c_n and h_j. With this relationship, we may rewrite Eqn.(2.11) as

$$y(k) = q_0 a(k) + \sum_{\substack{j=-\infty \\ j\neq k}}^{\infty} a(j) q_{k-j} + \sum_{n=-\infty}^{\infty} c_n \eta(k-n) \tag{2.13}$$

where the first term is a scaled version of the transmitted symbol, the second term is the intersymbol interference and the third term is the noise at the output of the equalizer. For convenience, we can set $q_0=1$. Consequently, if we could eliminate the second term (ISI) and the third term (noise) in Eqn.(2.13), the first term would tell us the exact value of the transmitted symbol.

For the moment, we will concentrate on the second term. The worst case ISI at the output of the equalizer is called the peak distortion D. This is given by

$$D = \sum_{\substack{j=-\infty \\ j\neq 0}}^{\infty} |q_j| = \sum_{\substack{j=-\infty \\ j\neq 0}}^{\infty} \left| \sum_{n=-\infty}^{\infty} c_n h_{j-n} \right| \tag{2.14}$$

where we note that D is a function of the taps c_n. If we choose to minimize this measure of error, we will be using the peak distortion criterion to find our optimum set of tap values. From Eqn.(2.14), we

note that D will be minimized (in fact, D will equal zero resulting in no ISI) if we choose the equalizer's taps so that the following condition will be satisfied

$$q_j = \sum_{n=-\infty}^{\infty} c_n h_{j-n} = \begin{cases} 1 & j=0 \\ 0 & j\neq 0 \end{cases} \tag{2.15}$$

By taking the z transform of this equation, we obtain

$$Q(z)=C(z)H(z)=1 \qquad C(z)=\frac{1}{H(z)} \tag{2.16}$$

where C(z) is the transfer function of the equalizer. Consequently, if we adjust the taps c_n in a manner which will make C(z) equal to the inverse of H(z), the sampled transfer function of the equivalent channel, there will be no ISI at the output of the equalizer. We see from Eqn.(2.15), that in the time domain, this has the effect of forcing all of the samples of the combined channel and equalizer impulse response to zero except for the cursor sample. From this, the algorithm which implements the peak distortion criterion gets the name, the zero-forcing (ZF) adaptation algorithm. This is a recursive algorithm which updates the value of each tap once per symbol period. The ZF adaptation algorithm is given by [17]

$$c_n(k+1)=c_n(k)+\mu e(k)a^*(n-k) \tag{2.17}$$

where $c_n(k)$ is the value of the nth tap at time t=kT, μ is the step size which controls the rate of adjustment, e(k) is the error signal at time t=kT and * denotes complex conjugate. The error signal is given by

$$e(k)=a(k)-y(k) \tag{2.18}$$

where a(k) will be either a known transmitted symbol or an estimate of the transmitted symbol made by the decision making device at the receiver.

The performance of a ZF equalizer (one that uses the ZF algorithm) can be expressed in terms of the noise power at the output. If the channel suffers from additive white Gaussian noise with power spectral density N_0, then the total noise power at the output of the equalizer is given by [18]

$$\sigma^2(ZF) = 2N_0T \int_{-1/2T}^{1/2T} \frac{df}{H(e^{j2\pi fT})} \tag{2.19}$$

where

$$H(e^{j2\pi fT}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} |F(f+m/T)|^2 \tag{2.20}$$

is the folded spectrum of the received pulse and f(t) is the shape of the received pulse before matched filtering. We immediately see that if there is a deep notch or even worse, a zero, in the folded frequency response of a channel, the noise power will be very large or infinitely large, respectively. This is because the equalizer will try to provide enough gain to flatten the response of the channel and will amplify any noise in the band of frequencies where the notch or zero occurs. Consequently, the biggest problem with a ZF equalizer, and hence the ZF algorithm, is that it can cause a large amount of noise enhancement. This stems from the fact that a ZF equalizer is only interested in eliminating ISI and completely ignores the noise term in Eqn.(2.13).

2.1.3.3.2 Mean-Square-Error (MSE) Criterion and the Least Mean Square (LMS) Adaptation Algorithm

If the MSE criterion is used to adjust an equalizer's taps, the coefficients are chosen to minimize the mean-squared value of the error

given in Eqn.(2.18). If we substitute Eqn.(2.13) into this expression, we find that this error signal can be expressed as

$$e(k)=a(k)-y(k) = \sum_{\substack{j=-\infty \\ j\neq k}}^{\infty} a(j)q_{k-j} + \sum_{n=-\infty}^{\infty} c_n\eta(k-n) \tag{2.21}$$

We immediately note that this measure of error includes both the ISI and noise terms from Eqn.(2.13). This is significantly different than the measure of error used by the peak distortion criterion which only includes, and tries to minimize, the ISI term.

It can be shown that the transfer function of a linear equalizer based on the MSE criterion is [18]

$$C(z)= \frac{1}{H(z)+2N_0/\sigma_A^2} \tag{2.22}$$

where N_0 is the power spectral density of an additive white Gaussian noise and σ_A^2 is the variance of the data symbols a_k. From this expression, it can be shown that the noise power at the output of the equalizer is given by [18]

$$\sigma^2(MSE)=2N_0T \int_{-1/2T}^{1/2T} \frac{df}{H(e^{j2\pi fT})+2N_0/\sigma_A^2} \tag{2.23}$$

If we now compare Eqn.(2.23) to Eqn.(2.19), we see that the noise power at the output of an equalizer based on the MSE criterion will always be smaller that the noise power at the output of an equalizer based on the ZF criterion. This is due to the extra positive term in the denominator of Eqn.(2.23) which will always make this integrand smaller. Consequently, a MSE equalizer will provide a compromise where some small residual ISI is allowed to remain after equalization, but the amount of

noise enhancement is reduced. This will result in a smaller probability of error. Due to the smaller noise enhancement caused by a MSE equalizer, the MSE criterion will be used to find the equalizer coefficients in this simulation study.

The algorithm which implements the MSE criterion is called the least mean-square (LMS) adaptation algorithm. We can derive this algorithm by first considering a finite length linear transversal filter with a total of N=N1+N2+1 taps. If the equalizer has input samples x(k) and tap coefficients c_n, then its output y(k) is given by the convolution

$$y(k)=\sum_{n=-N1}^{N2} c_n x(k-n) \tag{2.24}$$

This equation can also be written in vector form, which results in

$$y(k)=C^t X(k) \tag{2.25}$$

where

$$X(k)=[x(k+N1),x(k+N1-1),\ldots,x(k-N2)]^t$$

and

$$C^t=[c_{-N1},c_{-N1+1},\ldots,c_{N2}]$$

are the input vector and tap coefficient vector, respectively. The ^{t} superscript denotes transposition. The error signal e(k) can also be written in vector form as

$$e(k)=a(k)-y(k)=a(k)-C^t X(k). \tag{2.26}$$

It should be noted, that in general, the samples x(k), y(k) and a(k) are all complex valued. Consequently, in general, the taps c_n will also be complex valued. We may now define the performance index for the MSE criterion which is simply the mean-squared error J_{ms} given by

$$J_{ms}=E(|e(k)|^2)=E(|a(k)-y(k)|^2) \tag{2.27}$$

where $E\{*\}$ denotes the expectation operator. This can be expanded, using $y(k)=X^t(k)C$, to get

$$J_{ms}=E\{|a(k)|^2\}-E\{a(k)y^*(k)\}-E\{a^*(k)y(k)\}+E\{|y(k)|^2\}$$
$$= \sigma_A^2 - 2C^tRe[P] + C^tRC^* \qquad (2.28)$$

where

$\sigma_A^2=E\{|a(k)|^2\}$ is the average power of a(k),

$R=E\{X(k)X^h(k)\}$ is the autocorrelation matrix of x(k),

$P=E\{a^*(k)X(k)\}$ is the cross-correlation vector of a(k) and x(k),

and ^{h} denotes the conjugate transpose. Now, to minimize the MSE, we must take the gradient of J_{ms} with respect to the coefficients C and then set the gradient to zero to find the optimum set of coefficients C°. For the moment, we will simplify Eqn.(2.28) by assuming a real-valued process. This leads to

$$J_{ms}=\sigma_A^2-2C^tP+C^tRC \qquad (2.29)$$

Setting the gradient to zero results in

$$\nabla_c J_{ms}=\nabla_c\sigma_A^2-2\nabla_c(C^tP)+\nabla_c(C^tRC)=0$$
$$=-2P+2RC=0$$
$$RC=P \qquad (2.30)$$

where ∇_c denotes the gradient with respect to C. Consequently, a unique solution for the optimum set of equalizer taps is given by

$$C^\circ=R^{-1}P \qquad (2.31)$$

Having this expression there are two ways of solving for C°. There is the brute force method where we compute R and P, invert R and then multiply by P. For equalizers with more than a few taps this would be very computationally demanding and would require a great deal of hardware.

The second way of solving this problem is by finding an iterative solution. This method sacrifices some final accuracy for large savings in complexity. If we look back at Eqn.(2.28), we see that J_{ms}, the mean-squared error, is a quadratic function of the tap coefficient vector C. Thus, we can imagine the MSE as a N-dimensional punch bowl with a definite bottom or minimum. The minimum MSE corresponds to the coefficients C°. What is desired, then, is an algorithm that will adjust the coefficients, in such a manner, that they can start at any initial values and will gradually move to their optimum values corresponding to the bottom of the bowl. One technique, used to do this, is to adjust each tap gain in a direction opposite to the gradient of the MSE with respect to that tap's gain. In this way the MSE will always move closer to its minimum. For instance, the gradient of the MSE for a single tap is given by

$$dJ/dc \simeq \Delta J/\Delta c. \tag{2.32}$$

where ΔJ is the change in the MSE and Δc is the change in the value of the tap. Each time the tap is updated, we desire that the MSE decrease. This implies that ΔJ is always negative. By examining Eqn.(2.32), knowing ΔJ is a negative number, it is obvious that the sign of Δc must always be opposite to the sign of the evaluated gradient. This can be written in an equation as follows

$$C(k+1)=C(k)-\mu\nabla_c J_{ms}\Big|_{C=C(k)} \tag{2.33}$$

where μ is a small positive constant. Now, when the gradient is positive, the individual taps in C(k+1) will be smaller than the individual taps in C(k) and when the gradient is negative, the taps in C(k+1) will be larger than the taps in C(k). Both of these situations

will cause J_{ms} to be reduced and thus brought closer to its minimum value.

There is one last change to make to Eqn.(2.33) before the derivation is complete. The true gradient of the MSE is again complicated because of the need to find the auto- and cross-correlation matrices and then to multiply them together. A much simpler estimate of the true gradient can be found by taking the gradient of the instantaneous squared error. Thus, instead of the true gradient

$$\nabla_c J_{ms} = \nabla_c E\{e^2(k)\}$$

use the gradient of the instantaneous squared error

$$\begin{aligned} \nabla_c e^2(k) &= 2e(k)\nabla_c e(k) \\ &= -2e(k)\nabla_c(a(k) - C^t X(k)) \\ &= -2e(k)X(k) \end{aligned} \tag{2.34}$$

We may now substitute the estimate of the gradient for the true gradient in Eqn.(2.33) to get the LMS (least mean-squares) adaptation algorithm for real-valued processes.

$$C(k+1) = C(k) + \mu e(k)X(k)$$

or

$$c_n(k+1) = c_n(k) + \mu e(k)x(k-n) \qquad n=-N1,-N1+1,\ldots,N2 \tag{2.35}$$

where $c_n(k)$ is the nth tap at time k. For complex valued processes the result is very similar

$$C(k+1) = C(k) + \mu e(k)X^*(k)$$

or

$$c_n(k+1) = c_n(k) + \mu e(k)x^*(k-n) \qquad n=-N1,-N1+1,\ldots,N2 \tag{2.36}$$

where only μ, the step size, is real valued. This derivation was done for a linear transversal filter. We will now see that the update formula for a DFE is very similiar.

Consider a DFE with a forward filter having N1+N2+1 taps and a feedback filter with N3 taps. The equalizer's output can be written as

$$y(k)=\sum_{n=-N1}^{N2} c_n x(k-n) - \sum_{m=1}^{N3} b_m d(k-m) \qquad (2.37)$$

where b_m are again the feedback filter's coefficients. This equation can be written in a vector form to give

$$y(k)=C^t X(k)-B^t D(k) \qquad (2.38)$$

where $C^t=(c_{-N1},c_{-N1+1},\ldots,c_{N2})$

$X(k)=[x(k+N1),x(k+N1-1),\ldots,x(k-N2)]^t$

$B^t=(b_1,b_2,\ldots,b_{N3})$

$D(k)=[d(k-1),d(k-2),\ldots,d(k-N3)]^t$

This equation can be further simplified by combining the two coefficient vectors into one single vector and by combining the two input vectors into a single vector as follows, let

$CB^t=(C^t,-B^t)$ and $XD(k)=[X(k),D(k)]^t$

Thus, the equalizer's output can be written as

$$y(k)=CB^t XD(k) \qquad (2.39)$$

The error signal can now be obtained in the same manner as was done for a linear equalizer, except a different substitution is made for y(k)

$$e(k)=d(k)-y(k)=d(k)-CB^t XD(k) \qquad (2.40)$$

We now observe that Eqns.(2.25) and (2.26) are very similiar to Eqns.(2.39) and (2.40), respectively. Consequently, we can use the previous results to jump straight to the LMS adaptation algorithm for a DFE.

$$CB(k+1)=CB(k)+\mu e(k)XD^*(k) \qquad (2.41)$$

which can be broken down into two parts,

$$c_n(k+1)=c_n(k)+\mu e(k)x^*(k-n) \qquad n=-N1,-N1+1,\ldots,N2 \qquad (2.42)$$

for the forward equalizer and

$$b_m(k+1)=b_m(k)-\mu e(k)d^*(k-m) \qquad m=1,2,\ldots,N3 \qquad (2.43)$$

for the feedback filter. The indices n and m denote the tap positions. These equations are typically adapted jointly and once per a symbol interval.

The convergence properties and characteristics of the LMS algorithm are thoroughly discussed in [19] and [20]. Of all the properties analyzed, one of the most important is the influence the step size μ has on the convergence and the steady state error. When the step size is relatively small, the coefficients take a longer time to converge and are not able to track changes in the channel's characteristic as quickly. However, a smaller step size results in a final MSE which is closer to the minimum MSE. This can be seen by examining the expression for the steady state MSE which is given by

$$J_{ss} \simeq J_{min} + \mu J_{min} \sum_{n=1}^{N} \lambda_n \tag{2.44}$$

where J_{min} is the minimum MSE obtained for the optimum set of equalizer taps C^0, μ is the step size and λ_n are the eigenvalues of the autocorrelation matrix R. These convergence and steady state error characteristics come as a direct result of the algorithm being iterative. When the taps and the MSE are close to their optimum and minimum values, respectively, they will wander around these values without actually equaling them. If the step size is chosen to be big, this wandering will be over a wider range and thus the steady state error will be relatively large. One way of getting the best of both worlds is to initially use a large step size for fast convergence and then reduce the step size when nearing the final values to get the benefit of a smaller MSE.

There should be one final observation made before leaving this section on adaptive equalization. Before normal data transmission begins the equalizer is trained by a known signal as shown in Figure 2.13. The training sequence which is known to the receiver, is sent by the transmitter, so that the equalizer will have near optimum coefficients when the true data transmission begins. Thus, initially, this known sequence is used to generate the error signal for the LMS algorithm. After this one time training period, the equalizer coefficients continue to adjust, but in a so called decision-directed manner. In this mode, the error signal is based on the receiver's decisions, which approximate the transmitted data. As long as the decisions are correct with a high probability, the adaptive equalizer will be able to reduce or eliminate the ISI caused by a digital subscriber loop without regard to its configuration.

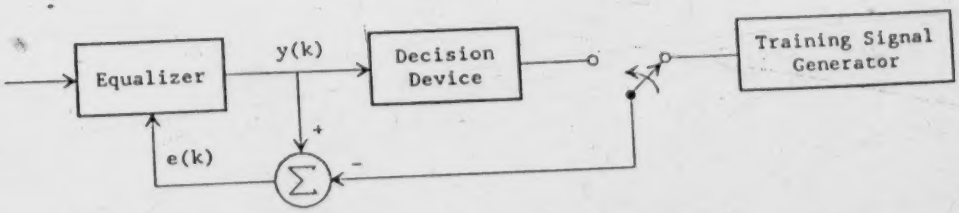


Figure 2.13 General structure of an adaptive equalizer

2.2 Transmission Disturbances

In addition to the problem of ISI there are a number of noise sources that impair the performance of a digital subscriber loop transmission system. These noise signals include echo, crosstalk, white noise and impulse noise. Depending on the configuration of the loop one or several of these noise signals will be the dominant source(s) of noise and can cause a serious degradation to the BER or the transmission rate achievable on a given digital subscriber loop. Thus, each of the noise sources will be examined to discover their cause and their relative strengths.

2.2.1 Echo

Echo comes as a result of an impedance mismatch between the balancing network of the hybrid circuit and the input impedance of the subscriber line. As was stated in the introductory chapter, the EC method of providing full duplex transmission requires that the hybrid circuit, as shown in Figure 1.4, separates the near-end (local) transmitted signal from the near-end (local) received signal and performs a two wire to four wire conversion. If within the hybrid, the balancing impedance Z_b is perfectly matched to the line impedance Z_ℓ, at every frequency, then there will be complete isolation between the transmitted and received signals. In practice, however, the balancing impedance is chosen as a compromise match to all possible line impedances. Consequently, there is typically a great deal of mismatch. This is especially true at low frequencies where the cable's impedance can have large variations with small changes in frequency. As a result, a significant part of the transmitted signal will leak through the

hybrid and be added to the received signal. This leaked portion of the transmitted signal is called local echo. Local echo can be the dominant source of noise if an echo canceller is not used to reduce it. For instance, suppose the received signal has passed through a digital subscriber loop which causes a 50 dB attenuation. The received signal's power will then be at -50 dB with respect to a 0 dB reference. The near-end transmitted signal only passes through the hybrid and thus only suffers the trans-hybrid loss. This loss is given by

$$\text{Trans-hybrid loss} = 20 \cdot \log_{10} \left| \frac{(Z_\ell + Z_s)(Z_b + Z_s)}{Z_s Z_b - Z_s Z_\ell} \right| \qquad (2.45)$$

which is typically 10 dB. Thus, the leaked signal or echo noise will be at approximately -10 dB with respect to the 0 dB reference. These numbers result in the noise power being 40 dB or 10,000 times stronger than our received signal power! To have a 20 dB signal to noise ratio (SNR), which is required to have near error free transmission, the echo canceller will have to reduce the echo noise by 60 dB. As can be seen from this calculation, an echo canceller has some demanding performance objectives.

At this point a discussion on what an echo canceller is and does is appropriate. We recall that the source of the echo is the near-end (local) transmitted signal. An echo canceller uses this known and available signal to generate a replica of the echo and then subtracts this replica from the received signal to cancel the disturbance. This is the same idea used by the ISI canceller to remove ISI in the previous discussion on equalization.

The echo canceller's output will typically be digital, but the echo cancellation can be done in either the analog or digital domain.

The difference is simply whether to convert the output of the echo canceller to analog using a D/A converter or convert the output of the hybrid to digital using a A/D converter. In either case the precision of the data converters (A/D or D/A) should be at least 12 bits for a 60 dB cancellation accuracy. In addition to this choice, there exist both linear and non-linear echo cancellers which results in a trade-off between poorer performance and added complexity.

2.2.1.1 Linear Echo Canceller

A linear echo canceller can be implemented with an adaptive transversal filter. The canceller linearly weights previously transmitted data symbols and then sums these values to produce a replica of the echo noise sample. The coefficients which weight the data symbols are adapted to match the impulse response of the echo's path and are updated using the previously derived LMS algorithm.

$$C(k+1)=C(k)+\mu e(k)A^{*}(k) \tag{2.46}$$

where μ is the step size, e(k) is the error signal and

$$C=(c_0,c_1,\ldots,c_{N-1})^{t}$$

$$A(k)=(a_k,a_{k-1},\ldots,a_{k-N+1})^{t}$$

are the complex canceller coefficient vector and the known complex transmitted data symbols vector, respectively.

Although the linear echo canceller is quite simple to implement, it has a number of problems. The first problem comes as a result of the likelihood that the echo canceller will be realized digitally and thus its coefficients will be quantized. Due to the large levels of cancellation that are demanded of the echo canceller, a coefficient word length of more than 20 bits is needed to ensure the quantization error

introduced by the digital tap values is below the 60 dB cancellation objective [1]. The second problem with a linear echo canceller is that it is not able to compensate for non-linearities in the echo path. Non-linearities can be introduced by the line driver and the data converters. In addition to these, there is typically some transmitted pulse asymmetry, which is another form of non-linearity, caused by the difficulty in generating positive and negative pulses that are identical within an accuracy of 60 dB except for sign. To solve these problems of non-linearity and precision, a non-linear echo canceller can be used.

2.2.1.2 Non-linear Echo Canceller

One of the most popular types of non-linear echo cancellers is the memory-based (look up table) echo canceller shown in Figure 2.14. The echo replica produced by this echo canceller can be expressed as

$$er(k)=f(a_k,a_{k-1},\ldots,a_{k-N+1})=f(A(k)) \qquad (2.47)$$

where f(*) is some general function (usually nonlinear) of the last N near-end transmitted symbols. If L different symbols can be transmitted, the echo canceller requires L^N words in its random access memory (RAM). The echo canceller works as follows. As each new symbol is transmitted its value is placed in a shift register, so that the values of the last N transmitted symbols are available at one time. This symbol sequence or data pattern held in the register is then used, along with some demultiplexers, to address a memory location within the RAM. At this memory location will be stored the echo replica associated with the data pattern or symbol sequence which addressed it. The output of the echo canceller is simply this value of the echo replica.

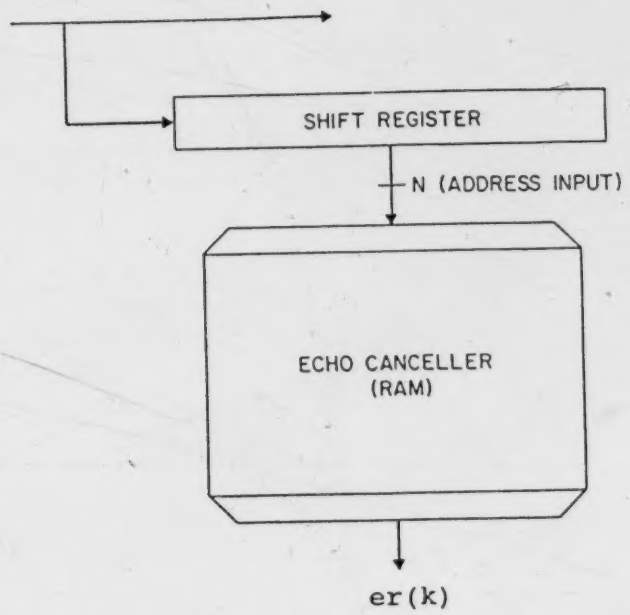


Figure 2.14 Non-linear echo canceller with one block of RAM

This echo canceller can be made adaptive by updating the memory location whose content is currently being used as the echo replica. The update equation is then

$$f(X,k+1)=\begin{cases} f(X,k)+\mu e(k) & \text{for } X=A(k) \\ f(X,k) & \text{for } X\neq A(k) \end{cases} \tag{2.48}$$

where X is the address of the memory location. It should be noted that while this echo canceller is being trained, the distant transmitter must be off. This ensures that e(k) will be determined properly.

The non-linear echo canceller has two obvious advantages. Its first advantage is that it can compensate for non-linearities in the echo path. Its second advantage is that it simply looks up the values for the echo replicas without doing any calculations. This saves on convolution operations. However, these advantages came with some disadvantages. The memory size of the non-linear echo canceller increases exponentially with L and N. This results in a very large memory. For example, if L=2 and N=16, then 65536 words of memory are required instead of only 16 words used by a linear echo canceller. This problem becomes overwhelming if L and N increase much beyond these values. In addition to this, the rate of convergence also slows down exponentially as L and N increase.

These memory and convergence problems can be partially eliminated by breaking the input data vector up into blocks and then performing non-linear echo cancellation on each block separately. Such an echo canceller is shown in Figure 2.15, where the echo canceller has M blocks. The echo replica is given by

$$\begin{aligned} er(k)=&f_1(a_k,a_{k-1},\ldots,a_{k-p+1})+f_2(a_{k-p},\ldots,a_{k-2p+1}) \\ &+\ldots+f_M(a_{k-(M-1)p},\ldots,a_{k-N+1}) \end{aligned} \tag{2.49}$$

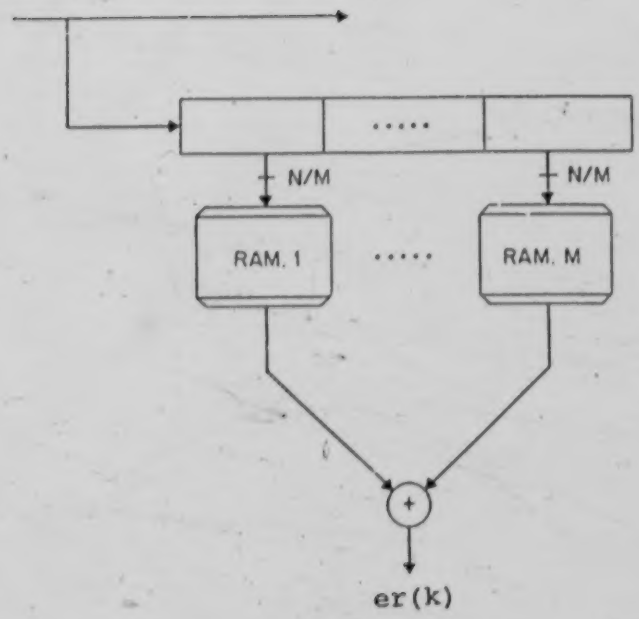


Figure 2.15 Non-linear echo canceller with M blocks of RAM

where p is the number of data symbols per a block and pM-N. The savings in memory can be seen by observing that each of the smaller RAMs will only require L^P words. This results in a total memory of $M*L^P$ words instead of L^N words. If we use the previous value for L-2 but break the canceller into M-2 blocks each having p-8 symbols (resulting in N-16), we find that only 512 words of RAM are required instead of the previous 65536 words. The rate of convergence also improves exponentially as the number of blocks, M, increases.

A non-linear echo canceller which uses two stages of cancellation was realized and tested in [21]. Its operation is somewhat different from what was described above, but provides an echo cancellation of 66 dB in addition to any cancellation provided by the hybrid. Thus, a 60 dB echo canceller is indeed realizable.

When cancellation of this magnitude is performed, the echo noise is typically reduced to a level where it is no longer significant. However, when there is an exceptionally bad mismatch between the impedances within the hybrid, echo can still be a noticeable impairment. These cases of bad impedance mismatching almost always occur on the subscriber (or network termination) side of the loop. This is due to the unpredictability as to the number, gauge, length and closeness of bridged taps to the subscriber's side. Thus, when an echo canceller is operating satisfactorily, echo is only a problem on the subscriber's side of the loop.

2.2.2 Impulse Noise

Impulse noise is the name given to infrequent high amplitude bursts of noise. It is generally evaluated by counting the number of

events that are above a certain threshold per a unit time. These bursts are predominantly caused by central office switching transients but, are also produced by lightning. It has been found that the number of impulse noise events is strongly dependent on the time of day. In a study summarized in [22], the greatest number of impulse noise events typically occurred in the periods between 9 and 11 a.m., 4 and 5 p.m. and 6 and 9 p.m. The first two periods would correspond to busy office times, while the third period would likely be calls made by individuals in the evening.

This study also examined the power spectrum of the impulse noise. It found that the spectrum was relatively flat, except for a gradual rise in power (about 10 dB) at frequencies below 40 kHz. At frequencies greater than 140 kHz the power spectral density appears to slowly taper off.

Since impulse noise is mostly caused by central office switching equipment, the subscriber's side of the loop is typically unaffected by this source of noise. However, the central office side can suffer severe degradations in performance because of high amplitude and frequently occurring noise bursts. This is especially true when considering older central offices which use electromechanical switching equipment.

Due to the nature of impulse noise, not much can be done to reduce its effects. However, when designing a system one important consideration is the choice of a line code. The line code should efficiently use as small a bandwidth as possible. This is necessary to bandlimit the signal and reduce the amount of noise entering the receiver. The code should at the same time try to have pulses with as

large an amplitude as possible for a given transmitted power. This will result in larger amplitudes at the receiver and thus the system will be less sensitive to impulsive noise bursts. However, these two goals are contradictory. A code can decrease its required bandwidth by having more levels, but this results in smaller amplitudes for the same transmitted power. It seems the best way of combating impulse noise is by replacing the old noisy electromechanical switches with new quieter digital switches.

2.2.3 White Noise

White noise is often the smallest contributor to the total noise in the system. However, it does have the effect of giving the system a noise floor, below which the total noise power cannot fall. White noise is characterized by having a zero mean and a flat or constant power spectrum over all frequencies. If the white noise is caused by a large number of sources then the central-limit theorem tells us that this noise will have a Gaussian distribution. This results in white Gaussian noise.

One source of white Gaussian noise is thermal or Johnson noise which is introduced by the resistors in the receiver. This noise is caused by random fluctuations in the amount of energy electrons in a conductor have. Thermal noise has a power spectrum which is constant to beyond a frequency of one terahertz.

Obviously, we do not want all of this noise power entering the receiver and being amplified by the equalizer. The system should again be designed to use its bandwidth as efficiently as possible. This will allow the noise outside the signal's bandwidth to be filtered and

consequently eliminated. This filtered noise is then called coloured Gaussian noise. This noise cannot be removed from our signal and simply degrades the system's performance.

2.2.4 Crosstalk

Crosstalk is, by far and wide, the largest and most important disturbance in the digital subscriber loop environment. It is usually the limiting impairment to high data rate transmission. Crosstalk is primarily caused by a signal being capacitively coupled from one wire pair to another. Its degrading effects are felt equally on both sides of the subscriber loop.

When the number of crosstalk interferers is large, crosstalk can be modeled by a Gaussian distribution. However, recently several authors [23],[24] have argued that crosstalk is caused by only a few wire pairs out of a bundle and as a result the Gaussian model is too conservative. These authors suggest that a more complex cyclostationary model for crosstalk be used.

There are two kinds of crosstalk on a digital subscriber loop, near-end crosstalk (NEXT) and far-end crosstalk (FEXT), as illustrated in Figure 2.16. NEXT represents the crosstalk of a local transmitter into a foreign local receiver. One model of its power transfer function is given by

$$|H_{NEXT}(f)|^2 = K_{NEXT} * |f|^{1.5} \quad . \tag{2.50}$$

where K_{NEXT} is a cable-dependent constant. In the ANSI basic access standards, it is suggested that NEXT models used in simulations should be calibrated to have a 57 dB loss at 80 kHz. This calibration results in a value of K_{NEXT}=88.179E-15 $Hz^{-3/2}$.

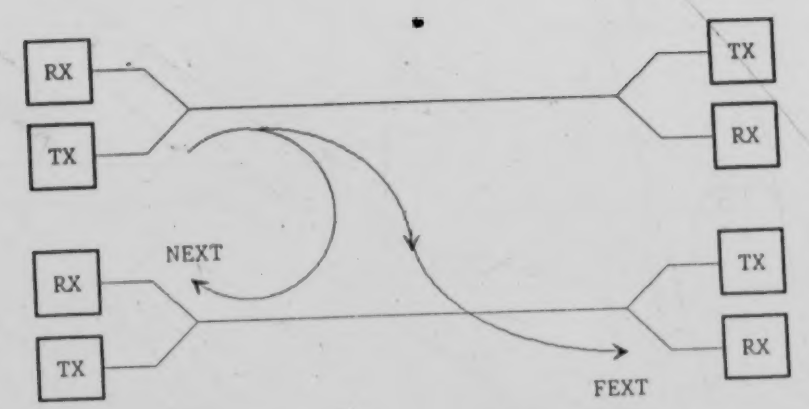


Figure 2.16 Illustration of NEXT and FEXT for two digital subscriber loops

FEXT represents the crosstalk of a local transmitter into a foreign remote receiver. Its equal level coupling loss (ELCL) power transfer function is modeled by

$$|H_{FEXT}(f)|^2 = K_{FEXT} * |f|^2 * L^{0.5} \qquad (2.51)$$

where K_{FEXT} is a cable dependent constant and L is the length of the cable. It must be noted, that in addition to this transfer function, the FEXT must pass through the subscriber loop with a transfer function F(f). Consequently, the FEXT suffers an additional loss from the cable which is given by $|F(f)|^2$, whereas, the NEXT does not.

The relative strengths of these two disturbances can be found by comparing the measured values of crosstalk given by Bell Laboratories [25]. These measurements were taken on 22 AWG cables which were bundled together in a 50 pair unit. At 3 MHz, the 99% worst case NEXT power which was coupled from 49 disturbing pairs into 1 disturbed pair was -38 dB. That is, the NEXT power in the 1 disturbed pair was at a level 38 dB below the power in each of the disturbing pairs. The corresponding value for the FEXT is -30.5 dB. This value is for a cable length of 300 m and does not include the additional loss which comes as a result of passing over the cable. By looking at these two values alone, the FEXT has the largest disturbing power and thus would seem to be the most significant disturbance. However, we are interested in frequencies which are considerably lower than 3 MHz and in loop lengths which are both longer and shorter than 300 m. Using Eqns.(2.50) and (2.51) and the measurements given above, we can find the following expressions for NEXT and FEXT at different frequencies and cable lengths.

$$NEXT(dB) = -38 - 15\log_{10}(3*10^6/f) \qquad (2.52)$$

$$FEXT(dB) = -30.5 - 20\log_{10}(3*10^6/f) + 5\log_{10}(L/300) \qquad (2.53)$$

where f is the frequency and L is the length of cable in metres. We again note, that the additional attenuation suffered by the FEXT from passing over the loop is not included. In this study, the relative strengths of these two disturbances at a frequency of 200 kHz would be of more interest. From the above equations, we find that the NEXT is at a level of -55.6 dB and the FEXT, for a 300m cable, is at a level of -54.0 dB. Now, the FEXT can be even greater for longer cables. Since the longest loop in the CSA is 3650 m we can use this value in Eqn.(2.53) to find the worst case FEXT. This value is -48.6 dB. By comparing this value to the NEXT power of -55.6 dB, we are again tempted to proclaim the FEXT the dominant disturbance. However, now comes the affect of the cable's attenuation. Since both the disturbed transmitted signal and the FEXT suffer the same attenuation from the cable, we can state that the signal to FEXT noise ratio, for this example, is equal to 48.6 dB regardless as to the actual value of the loss introduced by the cable. This signal to noise ratio is so good, that we can consider the FEXT negligible. However, the NEXT does not suffer this additional loss and hence, the signal to NEXT noise ratio will equal 55.6 dB minus the attenuation suffered by the disturbed signal. In some cases, this loss will be more than 35 dB, resulting in a signal to NEXT ratio of less than 20 dB. This is a far more serious disturbance. As a result of these calculations, NEXT is considered a dominant source of noise.

Due to the NEXT's increasing gain with increasing frequency (proportional to $f^{1.5}$), a system should again be designed to use its bandwidth as efficiently as possible. Any line codes and modulation techniques considered for use on a digital subscriber loop must ensure that the system is bandlimited and try to concentrate the signal's power

at lower frequencies. This combination will reduce the effects of NEXT noise.

Chapter 3

Transmission Techniques for the Digital Subscriber Loop

Now that an adequate knowledge of the transmission properties and noise sources in a digital subscriber loop system has been gained, we will examine some modulation and coding techniques that appear well suited to this environment. Hence, this chapter will begin with a discussion of modulation techniques which will include both baseband and passband transmission. Included in this discussion will be an introduction to some simple line codes which are used to increase the bandwidth efficiency of a system. In the later sections of the chapter, emphasis will be placed on expanded signal space coding techniques and the significant coding gains that they can provide.

3.1 Baseband Transmission Techniques with a Non-Redundant Line Code

One of the most common modulation techniques used in digital communications is baseband pulse amplitude modulation (PAM). A baseband PAM signal is generated by amplitude modulating a basic pulse by a sequence of real-valued data symbols. As a result, the information transmitted can be recovered by just examining the amplitude of the samples. A PAM signal can be expressed as

$$s(t)=\sum_{m=-\infty}^{\infty} a_m p(t-mT) \qquad (3.1)$$

where a_m are real-valued symbols, $1/T$ is the symbol or baud rate and $p(t)$ is the basic pulse.

The discrete values that a_m can assume are from a set of symbols called an alphabet. The size of the alphabet is determined by the

number of bits that are mapped to each symbol and by whether the alphabet has been expanded to introduce some redundancy into the system. Redundancy is usually introduced to correlate the transmitted bits so as to control some property of the signal. For now, the systems we will be considering have no redundancy. As a result, only the number of input bits per block will determine the size of the alphabet and thus determine the type of line encoding. If each block is made up of m bits, then the alphabet will contain $M=2^m$ symbols. The ANSI layer 1 specification committee has chosen the 2B1Q (2 binary bits to 1 quaternary symbol) line code as a standard for the basic access interface. Consequently, with m=2, the alphabet will contain M=4 symbols, each of which will represent one of four different transmission levels. The bit to symbol mapping is given in Table 3.1. Gray coding is used to ensure neighbouring symbols are only different by one bit. In this way if a symbol error is made at the receiver, it will likely result in only one of the two decoded bits being in error. Consequently, the bit error rate will be approximately one half the symbol error rate.

Input Block (First Bit, Second Bit)	Quaternary Symbol
00	-3
01	-1
11	+1
10	+3

Table 3.1 Bit to Symbol Mapping for 2B1Q Line Code

The average power spectrum of the signal s(t) in Eqn.(3.1) is given by [17]

$$S_s(f) = \frac{1}{T} |P(f)|^2 S_a(e^{j2\pi fT}) \tag{3.2}$$

where P(f) is the transfer function of the transmit filter (pulse shape) and $S_a(e^{j2\pi fT})$ is the average power spectrum of the symbol sequence a_m. $S_a(e^{j2\pi fT})$ is given by

$$S_a(e^{j2\pi fT}) = \sum_{n=-\infty}^{\infty} \phi_{aa}(n)\, e^{-j2\pi fnT} \tag{3.3}$$

where $\phi_{aa}(n)$ is the autocorrelation function of the symbol sequence a_m. Due to the input bits being uncorrelated after scrambling and since a non-redundant line coding is used, the sequence a_m will also be uncorrelated. Consequently, its average power spectrum is given by

$$S_a(e^{j2\pi fT}) = \phi_{aa}(0) = E[(a_m)^2] = \frac{1}{M} \sum_{i=1}^{M} (a_i)^2 = \sigma^2 \tag{3.4}$$

where σ^2 is the power or variance of the symbol sequence a_m and all the symbols in the alphabet are equiprobable. By substituting this result into Eqn.(3.2) we get

$$S_s(f) = \frac{\sigma^2}{T} |P(f)|^2. \tag{3.5}$$

Thus the shape of the power spectrum of the transmitted signal is solely determined by the energy spectrum of the basic pulse. If we assume raised cosine pulses will be used then the average power spectrum of the signal is given by

$$S_s(f) = \begin{cases} \sigma^2 T & 0 \le |f| \le (1-\alpha)/2T \\ \sigma^2 T/4\, [1-\sin(\pi T(f-1/2T)/\alpha)]^2 & (1-\alpha)/2T \le |f| \le (1+\alpha)/2T \\ 0. & |f| > (1+\alpha)/2T. \end{cases} \tag{3.6}$$

By noting that $S_s(f)$ is defined in terms of the magnitude of f, we realize that the average power spectrum is symmetric about d.c.. This is also true of the instantaneous power spectrum. These are fundamental characteristics of a real-valued baseband signal. In addition to these, we again find that the value chosen for α will play an important role in the amount of bandwidth needed by the signal.

Finally, in this type of digital communications system it is very important to maximize the bit rate achieved in a given channel bandwidth. This measure of performance is called the spectral efficiency of a modulation scheme. If there are M_b symbols in the alphabet and raised cosine pulses are transmitted, then the spectral efficiency of a baseband PAM signal is given by

$$\frac{\text{bit rate}}{\text{bandwidth}} = \frac{\frac{1}{T}\log_2 M_b}{\frac{1+\alpha}{2T}} = \frac{2\log_2 M_b}{1+\alpha} \text{ bits/sec/Hz} \qquad (3.7)$$

Consequently, the larger the value of M_b, the greater the spectral efficiency. From this expression alone, it would seem that a system designer should increase the number of levels or symbols transmitted without bound. Obviously, there is something missing. Most systems, as in a digital subscriber loop system, are constrained to transmit a certain average power. By increasing the number of levels M_b but maintaining a fixed average power, the distance between the transmission levels must decrease. This will result in a greater susceptibility to noise and will cause the symbol error rate to increase. Thus, there is a trade-off between larger spectral efficiency and smaller distances between transmission levels. When 100% excess bandwidth pulses are used

with the 2B1Q line code (M_b=4), the spectral efficiency is 2 bits/sec/Hz.

3.2 Passband Transmission Techniques with Non-Redundant Line Coding

Although there are many kinds of passband modulation schemes, we will concentrate on the type called quadrature amplitude modulation (QAM). A QAM signal is generated by modulating two carriers, that are 90° out of phase with each other, with two real-valued baseband PAM signals. This results in a signal which has both multiple amplitudes and multiple phases. It can be represented by

$$z(t)=s_r(t)\cos(2\pi f_c t)-s_i(t)\sin(2\pi f_c t)$$

$$=\left[\sum_{m=-\infty}^{\infty} ar_m p(t-mT)\right]\cos(2\pi f_c t)-\left[\sum_{m=-\infty}^{\infty} ai_m p(t-mT)\right]\sin(2\pi f_c t) \qquad (3.8)$$

where f_c is the carrier frequency, ar_m and ai_m are both real-valued sequences of symbols and $s_r(t)$ and $s_i(t)$ are real-valued baseband PAM signals which are known as the in-phase and quadrature components of $z(t)$, respectively.

This is one of several ways of representing a QAM signal. A second and more compact way of expressing this passband signal is made possible by introducing complex-valued signals. First, the two real valued PAM signals can be replaced by a single complex-valued baseband PAM signal $s_c(t)$ which is given by

$$s_c(t)=s_r(t)+js_i(t)=\sum_{m=-\infty}^{\infty} a_m p(t-mT) \qquad (3.9)$$

where the symbols $a_m=ar_m+jai_m$ are now taken from a complex-valued alphabet but the pulse shape $p(t)$ remains real-valued. This signal is

then modulated by a complex-valued carrier $e^{j2\pi f_c t}$ and finally, the real part is taken to give a second representation for a QAM signal.

$$z(t)=\mathrm{Re}(e^{j2\pi f_c t}\, s_c(t)) \tag{3.10}$$

When Eqn.(3.10) is expanded the result is equal to the representation given in Eqn.(3.8).

This second representation leads directly to the concept of using equivalent complex-valued baseband signals to represent real-valued passband signals. We note from Eqn.(3.10), that the QAM signal is generated by frequency translating the signal $s_c(t)$ up to the carrier frequency f_c. When this passband signal arrives at the receiver, it will have to be demodulated by the signal $e^{-j2\pi f_c t}$. This simply results in another frequency translation, but this time, the signal is translated back down to f=0. We could avoid these two frequency shifts, by simply leaving the transmitted signal as a complex-valued baseband signal and frequency translating the channel down to f=0. In fact, this is what has been done in this study.

The passband signal z(t) can be represented by its equivalent complex-valued baseband signal $s_c(t)$ which was previously defined in Eqn.(3.9). This complex-valued baseband signal has a Fourier transform $S_c(f)$ given by

$$S_c(f) = \int_{-\infty}^{\infty} s_c(t)\, e^{-j2\pi ft}\, dt \tag{3.11}$$

which may not be symmetric about f=0.

The transfer function of the channel H(f) also has an equivalent baseband representation which is given by

$$H_E(f) = H(f+f_c) \qquad f > -f_c \tag{3.12}$$

We see that $H_E(f)$ is obtained by simply shifting the passband transfer function down in frequency, so that instead of being centred at f_c, it is centred at f=0. This equivalent baseband channel transfer function $H_E(f)$ will have an impulse response $h_E(t)$ which, in general, will be complex-valued.

The response of a passband channel to a passband signal can now be found by using equivalent baseband representations. Suppose that a passband signal z(t) with an equivalent baseband signal $s_c(t)$ is transmitted on a passband channel with an impulse response h(t) and an equivalent baseband impulse response $h_E(t)$. The output of the system will also be a passband signal and can therefore be represented as

$$r(t)=\mathrm{Re}[e^{j2\pi f_c t}\, v(t)] \tag{3.13}$$

where v(t) is the equivalent baseband output signal. The passband output signal r(t) can be found from the convolution

$$r(t) = \int_{-\infty}^{\infty} z(\tau)\, h(t-\tau)\, d\tau \tag{3.14}$$

or equivalently, the output of the system, in the frequency domain is given by

$$R(f)=Z(f)H(f) \tag{3.15}$$

where Z(f) is the Fourier transform of z(t). From Eqn.(3.15), it can be shown that the equivalent baseband output of the system, in the frequency domain, is given by [17]

$$V(f)=S_c(f)H_E(f) \tag{3.16}$$

where all three terms in Eqn.(3.16) are equivalent baseband representations. This leads to the time domain relationship for the equivalent baseband output of a system which is given by

$$v(t) = \int_{-\infty}^{\infty} s_c(\tau)\, h_E(t-\tau)\, d\tau \qquad (3.17)$$

Consequently, through the use Eqns.(3.17) and (3.13), we can find the output of a passband system by using equivalent baseband representations.

From this knowledge, we can now model the whole passband system as an equivalent complex-valued baseband system. First, the transmitted signal will simply have a real and imaginary part as given in Eqn.(3.9) and will not have to be modulated by a carrier signal. Second, the passband channel will be shifted down in frequency to give the proper equivalent baseband channel as given in Eqn.(3.12). Finally, we can use the equivalent baseband response of the system v(t) as the received signal, without having to demodulate the passband signal with $e^{-j2\pi f_c t}$. In practice, however, the demodulated signal will have a phase offset which will have to be taken into account by the baseband system. The way this phase offset is incorporated into the equivalent baseband system is described in section 4.3.1 .

When a QAM signal is modelled as an equivalent complex-valued baseband signal, the complex-valued symbols used in the transmission are best displayed by plotting the alphabet as a set of points in a complex plane. Such a plot is called the signal constellation. Three examples of different signal constellations are shown in Figure 3.1. The first is for the 8 QAM or octal amplitude modulated phase modulated (AMPM) line code. The second is sometimes referred to as a square constellation and represents the 16 QAM signal . The third represents the 32 QAM signal and is called a cross constellation. Each of the points in these constellations have a one to one correspondence with

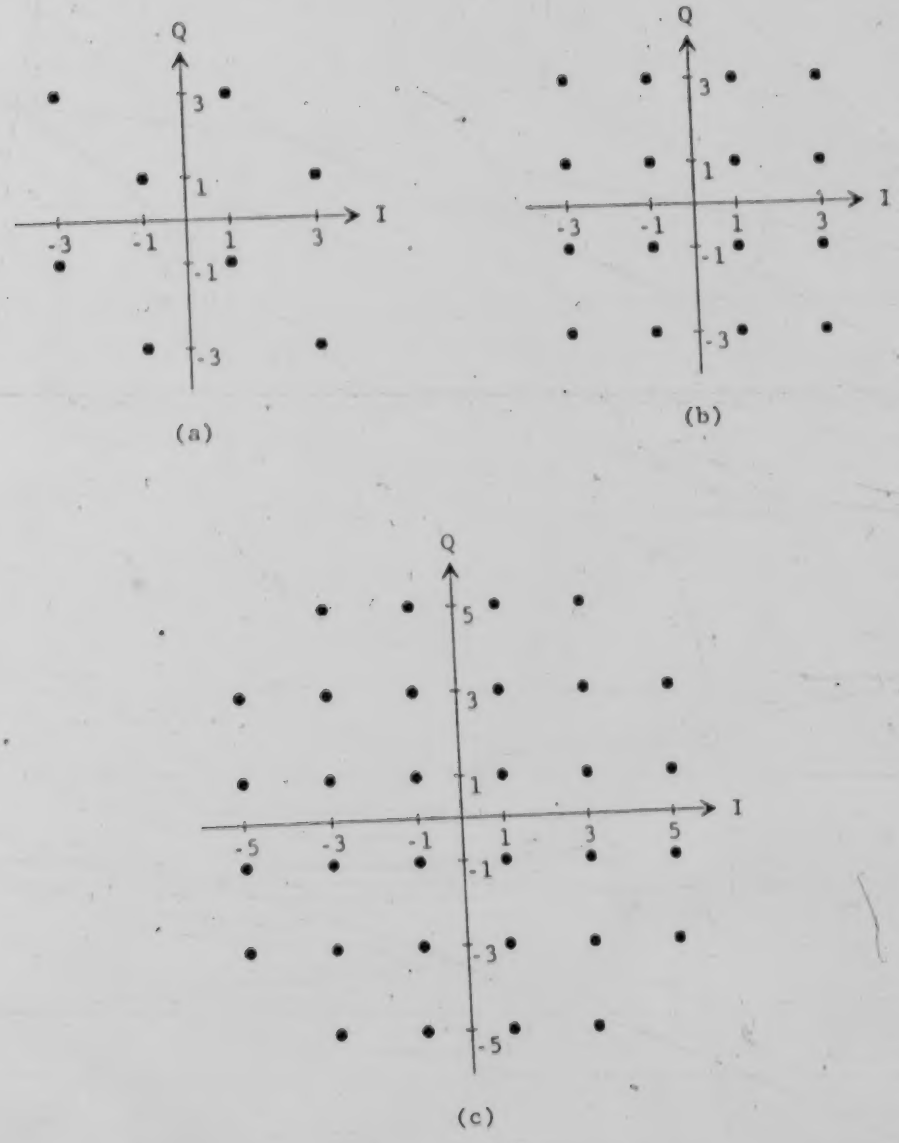


Figure 3.1 (a) Signal constellation for the 8 AMPM signal
(b) Signal constellation for the 16 QAM signal
(c) Signal constellation for the 32 QAM signal

both a symbol and a block of input bits. The average power associated with each of these constellations can be found through the relationship

$$\sigma^2 = \frac{1}{2} E[|a_m|^2] = \frac{1}{2M_Q} \sum_{i=1}^{M_Q} |a_i|^2 \qquad (3.18)$$

where M_Q is the number of symbols in the complex-valued constellation. Once again, it should be noted that when the transmitted signals have the same average power, the larger constellations will have smaller distances between adjacent points.

One of the most appealing features of QAM is its potentially good spectral efficiency. The spectral efficiency of a QAM signal using raised cosine pulses is given by

$$\frac{\text{bit rate}}{\text{bandwidth}} = \frac{\frac{1}{T}\log_2 M_Q}{2\,\frac{(1+\alpha)}{2T}} = \frac{\log_2 M_Q}{1+\alpha} \text{ bits/sec/Hz} \qquad (3.19)$$

By comparing Eqn.(3.19) to Eqn.(3.7), we note that a passband system must have $M_Q = M_b^2$ symbols in its constellation to have the same spectral efficiency as a baseband system with only M_b symbols in its constellation. This can be seen by observing that a passband system using 100% excess bandwidth pulses and the 16 QAM constellation has a spectral efficiency of 2 bits/sec/Hz which is the same result found previously for the baseband PAM signal using the 2B1Q line code. However, we should note that if a smaller value of α is chosen, such as 0.2, the spectral efficiency of the modulation scheme increases to 3.33 bits/sec/Hz. The choice of a smaller value for α can be justified in a passband system because of the fact that the channel's characteristic in the passband is typically much flatter than it is near d.c..

Consequently, the equalization will not be as difficult and a pulse shape with larger tails (smaller α) can be tolerated.

3.3 Transmission Techniques with Expanded Signal Space Coding.

The line coding techniques discussed up to now have not introduced any redundancy into the signal. When redundant bits or a controlled amount of correlation between data symbols is introduced, a sequence of input bits can be mapped to one of possibly several different symbols. This redundancy is often added to either control the shape of a signal's power spectrum or to reduce the number of decision errors at the receiver. The relative success a code has at achieving this latter goal is measured by its coding gain. A code's gain can be determined by operating a coded and an uncoded system at the same probability of error and then taking the difference between their SNRs (in dB). If a coded systems performance is better, it should require a smaller SNR than an uncoded system when both are operating at the same BER. Hence, there would be a coding gain.

Redundancy can be introduced into a signal in two ways. Suppose that a signal with no redundancy has an alphabet size of M_0 and a symbol rate B_0. These two values will define an information rate given by

$$I_0 = B_0 \cdot \log_2 M_0 \quad \text{bits/sec} \tag{3.20}$$

Since there is no redundancy, the transmitted bit rate will equal this information rate. However, when there is redundancy in a system, the transmitted bit rate will be greater than this information rate. One way of introducing redundancy and hence increasing the bit rate is to increase the symbol rate by choosing a $B > B_0$. This faster transmission rate provides opportunities to insert extra symbols, from the same

alphabet, which depend deterministically on the information bits. This method will increase the amount of bandwidth required by the signal which often leads to more noise entering the receiver. A useful code will have to more than compensate for this additional noise. These types of channel codes include block codes and convolutional codes.

The second way of transmitting redundancy in a signal is by increasing the number of symbols in the alphabet by choosing an $M > M_0$. This type of coding is called expanded signal space coding. It does not require a larger bandwidth than that of an uncoded system. Thus, this coding technique is very useful on bandlimited channels and on channels where extra bandwidth will causes an increase in the noise power entering the receiver. Such channels include digital subscriber loops. However, since the transmitted power cannot be increased, the symbols in the expanded constellation will have to be more closely packed than in an uncoded system. This will again lead to a reduction in the noise immunity, which a useful code will have to more than make up for.

These expanded signal space codes effectively combine the channel coding (convolutional or block codes) and modulation (bit to symbol mapping) operations into one. For the case where a convolutional encoder is used to provide the redundant bits needed by an expanded signal space code, the resulting code is called a trellis code. The means by which a trellis code is generated and decoded, as well as its performance (coding gain), will all be discussed in the following subsections.

3.3.1 Trellis Coded Modulation

The block diagram of a trellis encoder is shown in Figure 3.2. We can see that it is basically made up of two sub-blocks, a convolutional encoder and a signal mapper. The convolutional encoder takes in k of the m input bits and provides k+1 output bits. Thus, for every symbol period it generates one redundant bit which results in a total of m+1 bits being sent to the signal mapper. The reason a rate k/k+1 convolutional encoder is chosen stems from an observation made by Ungerboeck [26]. He observed that by simply doubling the size of the signal constellation almost all of the theoretically possible coding gain from signal set expansion can be achieved. Using more than twice the necessary symbols was shown to give considerably smaller returns. Thus instead of using $M=2^m$ symbols to send m bits as in an uncoded system, a trellis coded modulation (TCM) system uses 2M or 2^{m+1} symbols.

The way in which the signal mapper assigns its m+1 input bits to the 2^{m+1} symbols was also an important contribution by Ungerboeck. He proposed a mapping technique which maximized the free Euclidean distance between allowable symbol sequences. The free Euclidean distance is the minimum geometric distance between any two symbols or two sequences of symbols. His mapping method is known as mapping by set partitioning.

The first step in this procedure is to break the expanded constellation down into subsets. This must be done in such a manner as to ensure that the minimum Euclidean distance between symbols within subsets is maximized. An example of such a partitioning is shown in Figure 3.3. Here, an 8 PAM constellation, which is the result of expanding an uncoded 4 PAM (2B1Q) constellation, is shown with its

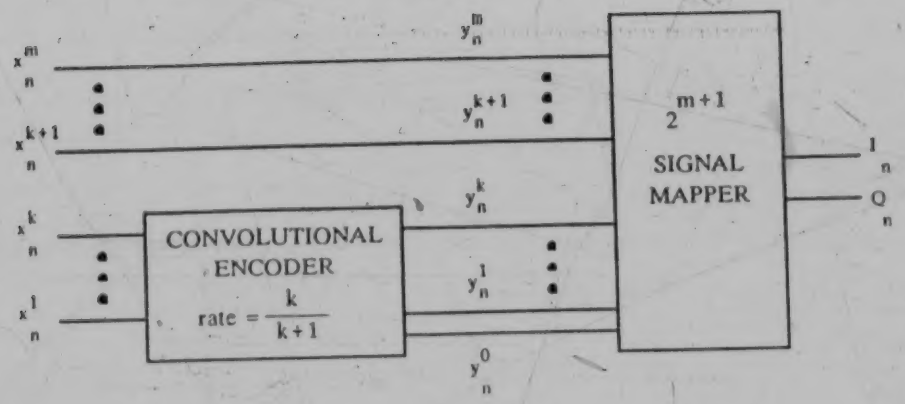


Figure 3.2 Block diagram of a trellis encoder

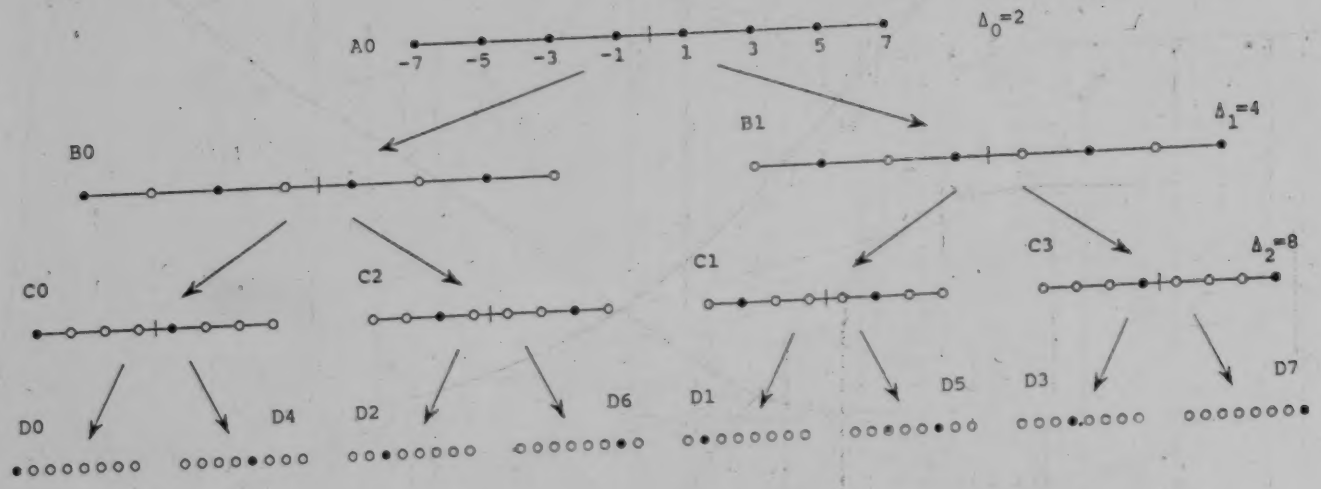


Figure 3.3 Set partitioning of the 8 PAM constellation

subsets. We see that as the constellation is continually broken down, the minimum distance between symbols within subsets is always maximized.

The second step in assigning these blocks of m+1 bits to symbols must make use of the convolutional encoder's trellis diagram. One stage of a trellis diagram shows encoder states and the different state transitions that are allowed. When several stages are placed next to each other, they provide a time axis which shows all the possible paths that an encoder can follow. An example showing one stage of a four state trellis diagram and the convolutional encoder which generates it is shown in Figure 3.4. The lines, representing transitions, connecting the nodes or states in the trellis are called branches. Each branch has input and output bits associated with it. This can be partially seen in Figure 3.4b where two of the output bits, y^1 and y^0, are shown directly above or below their respective branches. In this second step towards mapping bits to symbols, we must define which individual symbols from Figure 3.3 will be associated with which branches in the trellis. This will effectively map the symbols to the branches and since the branches are already associated with certain output bits, this will give us our desired bit to symbol mapping.

Ungerboeck has provided the following general rules to help assign channel symbols to trellis branches in a manner which maximizes the Euclidean distance between sequences of symbols.

1) First, if more than one branch connects the same two nodes (states) in a trellis diagram, these parallel transitions must be separated by the maximum possible Euclidean distance. Whenever there is a parallel transition, there exists the possibility of a single error event. The Euclidean distance associated with this

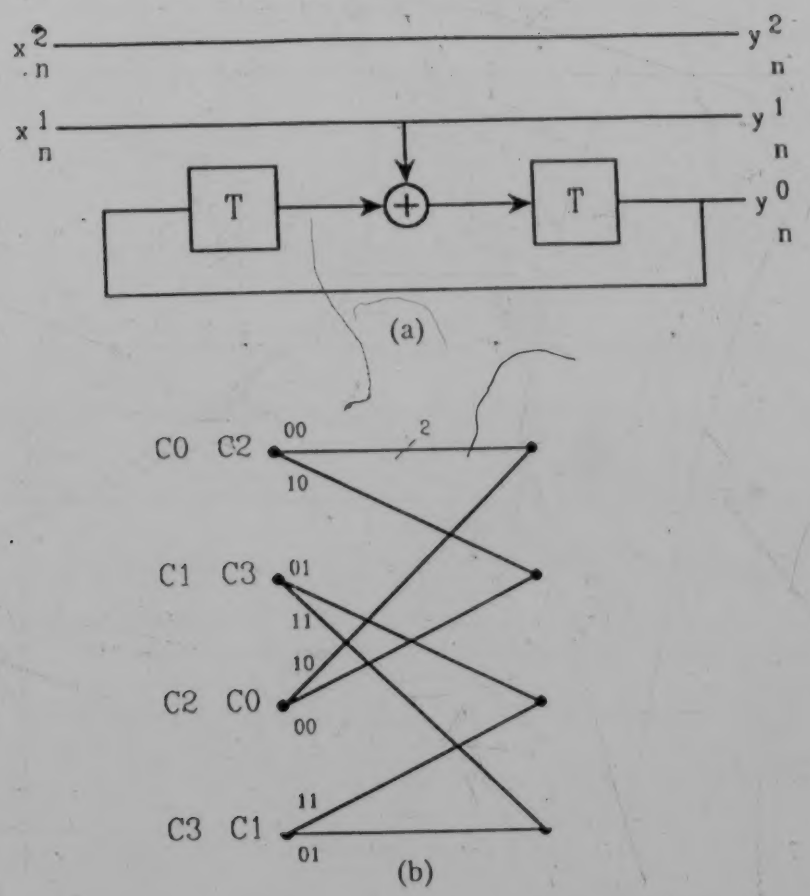


Figure 3.4 (a) 8 PAM four state convolutional encoder
(b) 8 PAM four state trellis diagram

single error event is the lower bound on the free Euclidean distance of the trellis code.

2) All transitions originating from or merging into the same trellis state should be assigned the next maximum Euclidean distance separation possible.

3) Use all symbols with equal frequency.

The best way of observing how these rules work is to apply them to the example in Figures 3.3 and 3.4. However, we should first note that trellis diagrams are often drawn showing only one branch connecting pairs of nodes when there are actually parallel branches present. This is the case in Figure 3.4b where each apparent branch in the trellis diagram actually represents two branches and thus two symbols. If the number of parallel branches is not illustrated, it can be deduced by examining the convolutional encoder. We observe that the input bit x^2_n is not coded as it passes directly through the convolutional encoder. This implies there will be 2^1 parallel transitions between each pair of connected states in the trellis. In general, if k bits are encoded, then m-k bits will pass directly through the convolutional encoder and will result in 2^{m-k} parallel transitions per a drawn branch. From the first rule, we must assign these transitions to symbols that are separated by the maximum Euclidean distance possible. As seen in Figure 3.3, the pairs of symbols which satisfy this rule are separated by a Euclidean distance of Δ_2=8 and are grouped in the subsets C0, C1, C2 and C3. Since at this point in the labeling either of the two symbols from within a subset could be assigned to either branch of a parallel transition, we may simply assign subset names to the drawn branches and leave the specific assignment of symbols to branches until later. When

the time comes, the actual symbol that will be transmitted will be determined by the value of the uncoded bit x^2_n.

The second rule tells us that branches entering or leaving the same state should be assigned to symbols which have next possible maximum Euclidean distance. This implies that drawn branches associated with a common node should be assigned to subsets that have the same parent subset. For instance, the subsets C0 and C2 have the same parent, B0, and as a result they should always be assigned to drawn branches that share a common node. This is also true of subsets C1 and C3. If this second rule is followed, all of branches entering and leaving the same node will be associated with symbols that come from either the parent subset B0 or B1 and will be separated by a Euclidean distance of at least $\Delta_1=4$. Thus, our task is almost complete. We must simply assign the subset names to drawn branches keeping the above groupings in mind and realizing that branches associated with the same output bits, obviously, must be assigned to the same symbol.

We may begin the mapping by assigning C0 (which could be any one of the four subsets) to the top branch leaving State 0. As a result of this, subset C2 must be assigned to the lower branch leaving the same state to satisfy rule 2. We should now observe which states these assigned branches enter so we can give subset assignments to the other branches entering these same states. For example, the first branch that we labeled C0 re-enters State 0. Consequently, the other branch entering this state will have to be assigned subset C2 to again satisfy rule 2. Likewise, the other branch entering State 1 must be assigned C0. Now, there are four branches that remain unassigned. To use all the symbols with equal frequency, the remaining branches will have to be

assigned with subsets C1 and C3. We may choose C1 (could use C3) to represent the top branch leaving State 1. The next three branches are then assigned in a similar manner to that described above. The final subset to branch assignment is as shown in Figure 3.4b. It should be stressed that this is just one of many possible configurations. This particular mapping does, however, result in the numerical part of the subset name (2 from C2) being the decimal equivalent of the binary number obtained by placing the two output bits, y^1_n and y^0_n, side by side. For example, a symbol from subset C2 will be transmitted when the output bits y^1_n and y^0_n are 1 and 0, respectively. This does not improve the code's performance but can make the bits to symbols assignments somewhat simpler.

The final bits to symbols mapping is shown in Figure 3.5. In general, this mapping is accomplished by identifying the output bits associated with a particular drawn branch, y^1_n and y^0_n in the example, and then assigning these bits to the symbol(s) associated with that branch. When parallel transitions exist, the different values that the uncoded bit(s), y^2_n in the example, can assume will select one of the symbols from all those belonging to the same subset. This will result in a unique one to one mapping of the convolutional encoder's output bits to the symbols in the expanded constellation. For example, the top branch leaving State 0 is associated with the output bits $y^1_n y^0_n$=00. Since we assigned the subset C0 to that branch, both of the symbols within this subset will be partially mapped by these two bit values. Their final bit mappings will then be made unique by letting one symbol be associated with y^2_n=0 and the other y^2_n=1. This is how trellis codes are designed using the mapping by set partitioning method.

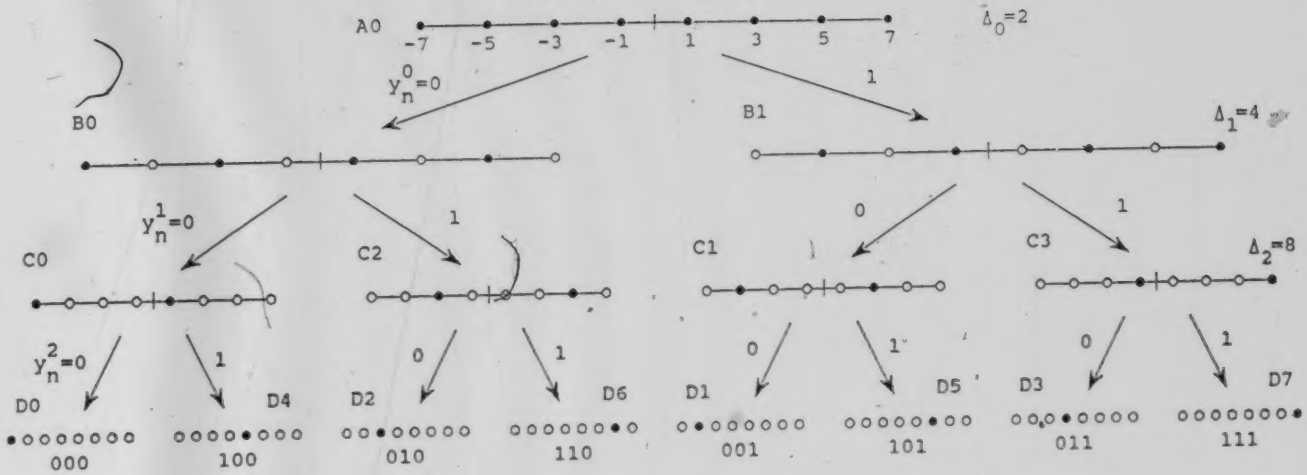


Figure 3.5 Set partitioning of the 8 PAM constellation showing the bits to symbols mapping

3.3.2 Viterbi Algorithm Decoder

Now that we are able to generate a trellis code, the next step is to find out how to decode it. One of the most common methods of decoding trellis codes is to implement an optimum soft-decision decoder using the Viterbi algorithm. A soft-decision decoder operates on unquantized output samples from the channel. It has a significantly better performance than a hard-decision decoder, which makes independent symbol decisions prior to decoding and results in an irreversible loss of potential information. In fact, the decision to use a soft-decision decoder was consciously made by Ungerboeck when he chose to maximize the free Euclidean distance of the codes, instead of the Hamming distance. This is seen by the fact that the most probable errors made by an optimum soft-decision decoder occur between symbols or sequences of symbols that are closest together in Euclidean distance.

The Viterbi algorithm (VA) is an optimum maximum likelihood sequence estimator. It achieves the minimum probability of error by finding the one sequence A'(k), from among the set of all possible transmitted sequences AA, which is closest in Euclidean distance to the received sequence Y(k). That is, it finds the sequence A'(k) which satisfies.

$$\sum_{k=-\infty}^{\infty} |Y(k)-A'(k)|^2 = \min_{(A(k))\epsilon AA} \sum_{k=-\infty}^{\infty} |Y(k)-A(k)|^2. \qquad (3.21)$$

The VA accomplishes this goal by having a full knowledge of the particular trellis diagram used to generate the code. In fact, a good understanding of how the algorithm works can be gained by thinking of a trellis diagram. Three stages of a trellis are shown in Figure 3.6. For the moment, we will concentrate on the kth stage. We recall from

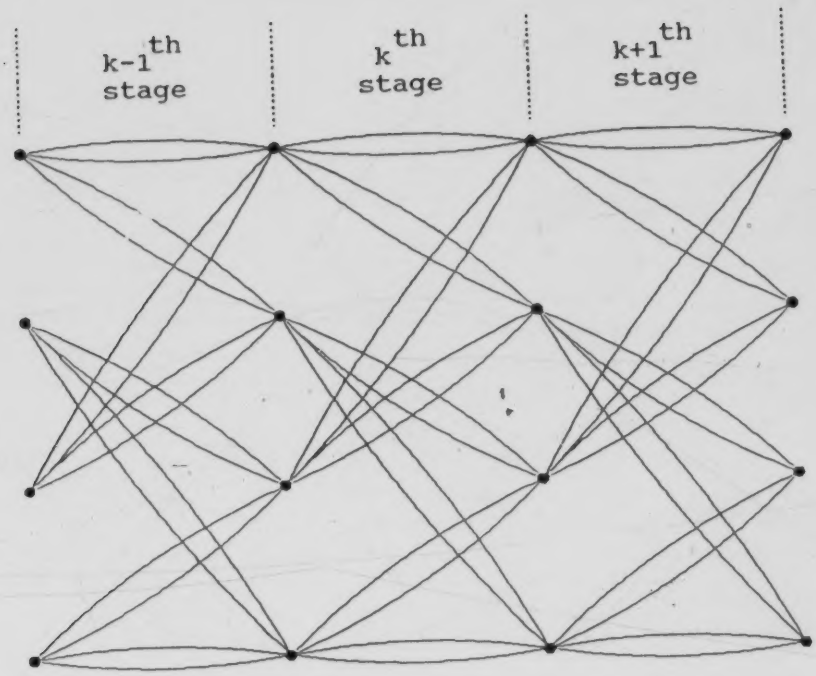


Figure 3.6 Three stages of a trellis diagram

the previous subsection, that each branch of a trellis is associated with a particular symbol, a_i. In addition to this, at time k, all of the branches in the kth stage of a trellis diagram will also be associated with a received sample, y_k. Consequently, once the value of y_k is known by the VA, the squared Euclidean distances between the symbols, a_i, and the received sample y_k, can be calculated and then assigned to their respective branches in the trellis. These distances are called branch metrics and for the kth stage are given by

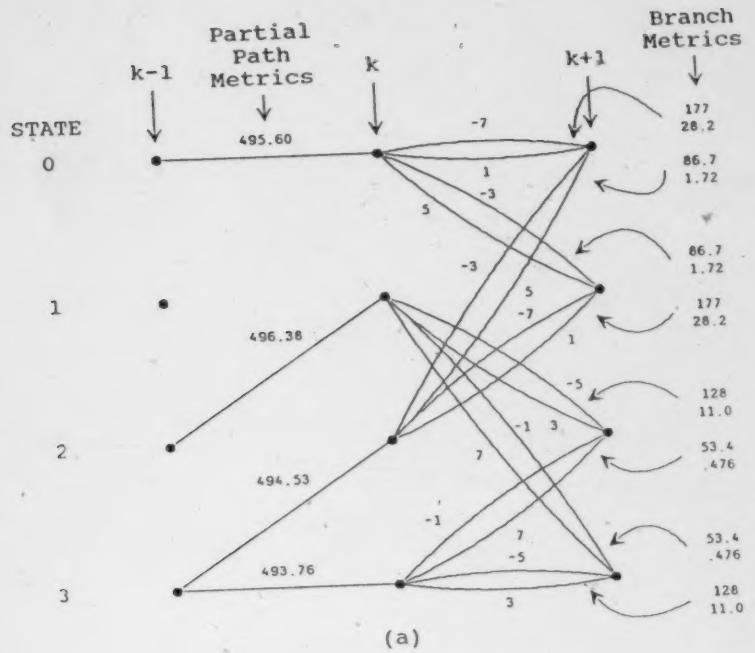
$$\text{branch metric}_{k,i} = |y_k - a_i|^2 \tag{3.22}$$

The closer the sample y_k is to a particular branch's associated symbol, the smaller the value of the branch metric.

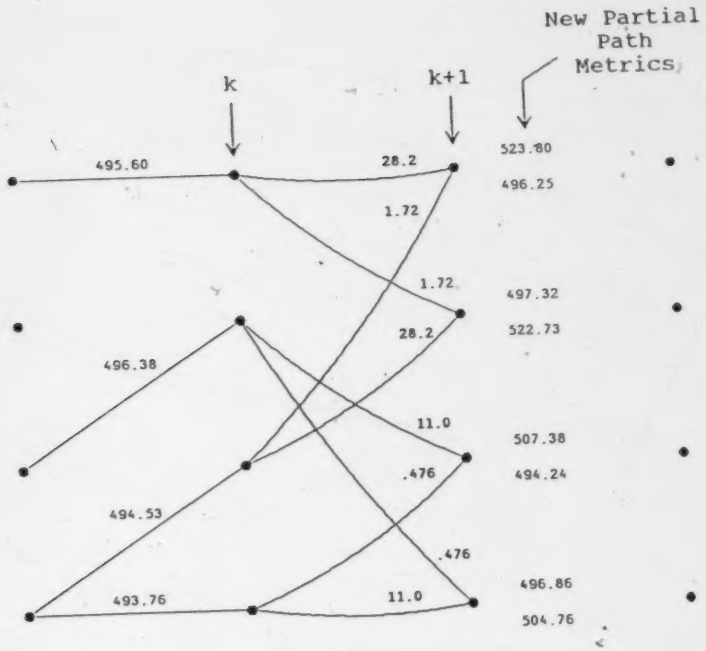
In addition to this, a total metric can be calculated for each path going through all the stages of a trellis diagram. This so called path metric, is the sum of all the component branch metrics which were used to build the path. The job of a maximum likelihood sequence estimator is to find the smallest of these path metrics. This, however, is no easy task since the total number of paths in a trellis grows exponentially with the number of stages in it. In fact, the brute force method of actually calculating every path metric possible is simply impractical. The VA, on the other hand, notes that as two or more branches converge to a single node, one of them will have a partial path metric (the sum of branch metrics to the left of this node) which is smaller than all the others. This observation permits the algorithm to throw away all of the larger partial path metrics because they are no longer candidates for the minimum total path metric. This early rejection of unlikely paths results in a substantial savings in decoder complexity.

The path having the smallest metric at each state is called a surviving path. Since we do not know which state the overall minimum path passes through at each stage of the trellis, we must keep one surviving path for each and every state. Fortunately, the surviving paths tend to merge into one path after a number of symbol periods. This enables the algorithm to make decisions on all the symbols which are older than this merging depth. Unfortunately, this depth is not deterministic and can fluctuate depending on the amount of noise present and the sequences transmitted. Consequently, a modification in the algorithm is usually made to force a decision after a delay of d symbol periods or trellis stages. When d, the truncation depth, is chosen large enough, this modification usually has negligible impact on the probability of error [18].

A brief example showing how the VA works will help clear up any loose ends. The example is shown in Figure 3.7 and again uses the 4 state trellis code described in the previous section. It begins with the configuration of Figure 3.7a where the path metrics of the surviving paths at a time just before k are as illustrated. At time k, an equalized sample, y_k=6.31 is received by the VA. It immediately calculates all the branch metrics, as listed beside the trellis, by using Eqn.(3.22). The VA now begins to eliminate unlikely paths, by first comparing the values of branch metrics involved in parallel transitions. It keeps only the branch with the smallest branch metric and discards the rest. For instance, the algorithm will throw away the branch associated with the symbol -7 having a metric of 177 and keep the branch associated with the symbol 1 having a metric of 28.2. Following this same idea for the rest of the parallel branches will result in the

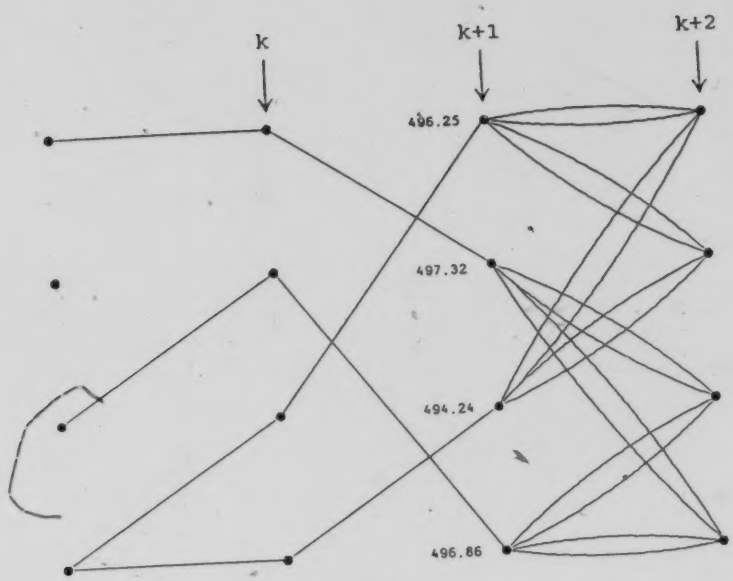


(a)



(b)

Figure 3.7 An example illustrating how the Viterbi decoder functions



(c)

Figure 3.7 An example illustrating how the Viterbi decoder functions (Con't)

remaining branches and metrics shown in Figure 3.7b. We now have only two branches entering each state instead of four. One of these two branches per state will now be eliminated by calculating a new partial path metric for each branch at time k+1 and throwing out the larger of the pair. For instance, we can calculate a new partial metric for each path entering state 0 at time k+1. The top branch has a metric of 28.2 which must be added to the old partial path metric of state 0 (495.60) since this is the previous state connected by the branch. The result is a path metric of 523.80. The lower branch comes from state 2 which has old partial path metric of 494.53. This must be added to the branch metric of 1.72 to give a result of 496.25. Since this path metric is smaller then the path metric calculated for the top branch, the lower branch will form the surviving path. The value of 496.25 will become the partial path metric of state 0 as shown in Figure 3.7c, along with the other survivors and their path metrics found in a similar manner. The algorithm has now finished processing this stage of the trellis and is waiting for another received sample. However, due to the delay in waiting for the surviving paths to merge, we will not know what symbol was transmitted at time k until the time k+d, where d is again the truncation depth.

3.3.3 Parallel Decision Feedback Decoding

In section 2.1.3.2.1 of chapter 2, the performance of a decision feedback equalizer (DFE) was found to be significantly better than those of either a symbol or fractionally spaced linear equalizer when the channel had severe amplitude distortion. This performance advantage came as a direct result of using past decision to immediately generate a

replica of the ISI and then subtracting this replica from the received signal to obtain ISI free samples. This scheme works extremely well for an uncoded system.

Unfortunately, its performance is not nearly as good in a trellis coded system where the equalizer is followed by a maximum likelihood sequence estimator such as the VA decoder. The reason for this is that the DFE requires that immediate decisions be made as to the value of the present received symbol so that it can be used to cancel ISI from the next sample it receives. This is in contrast to the operation of a Viterbi decoder which is a sequence detector. It may require possibly 10's of future samples before it concludes which symbol is associated with the present sample. Consequently, to satisfy the DFE, preliminary decisions must be made before the Viterbi decoder. However, it must be noted that these decisions will be coming from an expanded signal set which will have a relatively small Euclidean distance between adjacent symbols when compared to an uncoded system. This, together with the fact that the immediate decisions will ignore the intersymbol dependence introduced by the trellis code, leads to the conclusion that the preliminary decisions made before the decoder are more likely to be in error then decisions that would be made in an uncoded system. This larger probability of error in the preliminary decisions will cause a serious degradation in performance. [27][28]

One solution to this problem was proposed by Wesolowski [27]. He demonstrated that by putting the DFE function inside the Viterbi decoder a significant gain in performance was possible. Later in [29], this technique was given the name parallel decision feedback decoding (PDFD). Essentially, the distinctive feature of PDFD is that instead of using

only one sequence of decisions in the feedback path, equalization is accomplished by using a unique sequence of decisions for each state in the trellis. Where the feedback sequences are based on the history of each state's surviving path. As a result, instead of calculating branch metrics with one received sample per a trellis stage, there will be a unique decision feedback equalized sample for each state in the stage. This will result in an increase in the decoder complexity, but this increase is well justified by improved performance [29].

The PDFD is very similar to the VA decoder except for a couple of extra steps that come as a result of having to do the decision feedback equalization within the decoder. We will now discuss the PDFD in detail. We can begin at time k when the decoder receives a sample y_k from a forward equalizer. The first task is to provide decision feedback equalization to the sample. However, as stated above, a unique equalization is performed for each state in the trellis. Consequently, an ISI term which will be subtracted from y_k must be generated for each state. This is done by realizing that each of the surviving paths at time k is associated with a certain sequence of symbols. From this the ISI terms can be generated using

$$ISI_{s,k} = \sum_{m=1}^{N3} b_m d_{s,k-m} \qquad ,s=0,1,\ldots,S-1 \tag{3.23}$$

where S is the number of states, b_m are the DFE tap coefficients, N3 is the number of taps in the feedback filter and $d_{s,k-m}$ are the symbols associated with the surviving path from state s. As can be seen, this is the exact expression found in Eqn.(2.8) for the ISI, except we are now calculating one term for each state.

The next step taken by the decoder is to subtract these terms from the received sample y_k. This results in

$$V_{s,k}=y_k-ISI_{s,k} \qquad ,s=0,1,\ldots,S-1 \tag{3.24}$$

where $V_{s,k}$ are the decision feedback equalized samples at time k. This is the last step in the equalization of the signal.

What remains now is to simply decode the signal by using a slightly modified VA. The modification comes in the calculation of the branch metrics. Instead of comparing all the symbols to y_k, branches leaving state s must have their metrics calculated using the equalized sample $V_{s,k}$. That is, for the kth stage, the branch metrics are given by

$$\text{branch metric}_{k,s,i}=|V_{s,k}-a_i|^2 \qquad ,s=0,1,\ldots,S-1 \tag{3.25}$$

In this way all of the symbols associated with branches leaving state 0 will be compared to $V_{0,k}$ and all those symbols associated with branches leaving state 1 will be compared to $V_{1,k}$, etc.. Once these metrics are calculated, the PDFD uses the VA in exactly the same way as was described in the previous subsection.

3.3.4 Performance of Trellis Codes

The theoretically possible coding gain that a TCM system can provide over an uncoded system is relatively easy to find. Since there is no bandwidth expansion, the noise variance will be the same for both cases. Consequently, we only need to compare their free Euclidean distances to discover their relative performances. However, we must ensure that this comparison is done between systems sending the same number of bits per symbol and that the average signal powers in both

systems are taken into account. The coding gain G of a TCM system is given by

$$G=10\cdot\log_{10}\left[\frac{(d^2_{min}/P_{av})_{coded}}{(d^2_{min}/P_{av})_{uncoded}}\right] \quad dB \qquad (3.26)$$

where d_{min} is the free (minimum) Euclidean distance among all valid sequences and P_{av} is the average power of the signal.

To get an appreciation for how this gain is calculated, we will show an example using the 4 state trellis code previously designed. Figure 3.8a shows the 8 PAM constellation with its bit to symbol mappings. Conveniently, each symbol value also describes the Euclidean distance between itself and the centre of the constellation, in arbitrary units. The average signal power of this constellation can be calculated using

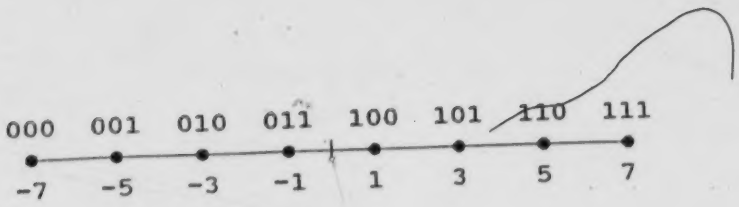
$$P_{av}=\frac{1}{M}\sum_{i=1}^{M}|a_i|^2 \qquad (3.27)$$

where a_i is the Euclidean distance of the ith symbol from the centre of the constellation and there are M symbols. Consequently, the 8 PAM constellation has an average signal power of $P_{av}=21$. The constellation of an uncoded 4 PAM signal which had its Euclidean distances chosen to produce the same average power as the 8 PAM constellation is shown in Figure 3.8b. Making the two average powers equivalent was not necessary but simply permits an obvious comparison of the uncoded and coded d_{min}.

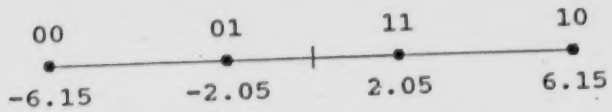
The free Euclidean distance of an uncoded signal is simply the smallest distance between symbols in the constellation. Consequently, d_{min} for this 4 PAM signal is given by

$$(d_{min})_{uncoded}=(6.15-2.05)=4.10$$

and $(d^2_{min})_{uncoded}=16.81$.



(a)



(b)

Figure 3.8 4 and 8 PAM constellations with P_{av}=21

The calculation of the free Euclidean distance of a TCM system is not quite as straight forward. We must find the so called minimum distance error event for the code and then calculate how far in Euclidean distance this error event is away from the correct symbol or sequence of symbols. This distance will be the value of $(d_{min})_{coded}$.

As was previously stated, the free Euclidean distance of a code is upper bounded by the distance between parallel branches. This is true since these branches imply single error events. For the trellis coded 8 PAM signal, the distance between parallel branches or transitions is $d_{par}=\Delta_2=8$.

Next, we must find the smallest Euclidean distance associated with a multiple error event. Once we find this value we can compare it to the value of d_{par} and then by choosing the smaller of the two, we will have the free Euclidean distance of the coded system. By examining the trellis stages in Figure 3.9, we note that double error events cannot occur (ignoring parallel branches). Thus, we shall begin by examining a triple error event such as the one shown by a bold line. Here, the top sequence will be assumed correct and corresponds to the all zero path (all transmitted bits were zero). Each of the symbols transmitted on this path have a Euclidean distance of -7 units from the centre of the constellation. The lower path is the error event and corresponds to having chosen the path with Euclidean distances of -3, -5 and -3 from the centre of the constellation. We can now find the Euclidean distance between these two paths to discover if it is a candidate to be $(d_{min})_{coded}$. Using the notation of Figure 3.3, we get

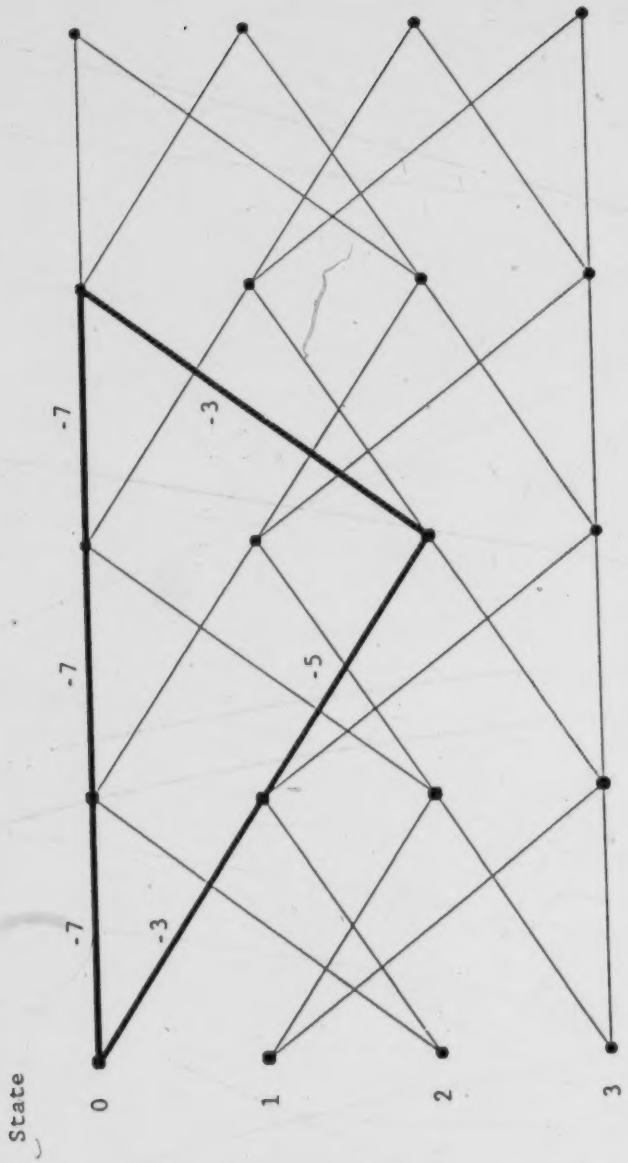


Figure 3.9 Illustration of a minimum distance error event

$$d^2_{bra3} = \Delta_1^2 + \Delta_0^2 + \Delta_1^2$$
$$= (-7+3)^2 + (-7+5)^2 + (-7+3)^2 = 36$$

$$\therefore d_{bra3} = 6$$

Indeed, it is smaller than d_{par}. However, is this now the minimum distance error event? The answer to this can be found by examining the trellis in Figure 3.9. We note that since there are no catastrophic error events, a sequence of errors will have to both leave and then rejoin the correct path. Thus, a multiple error event will have to be different from the correct path by at least two branches which are both a distance of Δ_1 away. We recall, however, that double error events are not possible. Thus at least a third branch will have to be in error. By examining the trellis we see that once departed from the correct path the error event must pass through a third branch which is different from the correct branch by a distance of Δ_0. By squaring these three distances and then summing them, we see that we get the same value as given for d^2_{bra3}. Now, if the error event passes through four or more branches, it will still have at least this minimum squared Euclidean distance. Consequently, $d_{bra3} = (d_{min})_{coded} = 6$ and $(d^2_{min})_{coded} = 36$.

By placing this value, along with the others previously found, into Eqn.(3.26) we get the following coding gain

$$G = 10 \cdot \log_{10}[(36/21)/(16.81/21)] = 10 \cdot \log_{10}(2.14)$$
$$= 3.31 \text{ dB}$$

Hence, even simple 4 state codes can give a fairly substantial coding gain. This coding gain can be increased by increasing the number of states in the trellis diagram. For example, an 8 state code typically provides a gain of 4 dB, while codes with gains as large as 6 dB are possible when using 128 or more states.

Now, a word of caution. These codes were designed to be optimum on ISI free additive white Gaussian noise channels. When either of these two criteria are not met, we can expect a code's performance to be somewhat poorer. In addition to this, these values are asymptotic coding gains. As such, they are only achievable at very high SNRs. Hence, when actually implemented, these codes may not provide their full asymptotic gains, but should still produce a significant performance advantage.

3.3.5 Rotationally Invariant Trellis Codes

Both the QAM and the two dimensional (complex signals) TCM signals require a coherent demodulator at the receiver. This means that the demodulating signal should have exactly the same carrier frequency and carrier phase as the incoming signal. In practice, however, the receiver and the transmitter will usually have their own crystal oscillators which will act as independent time bases. As a result, the carrier recovery circuit in the receiver will have to derive or estimate the carrier frequency and phase directly from the incoming signal. The effect this estimate will have on the system's performance can be calculated as follows. We may begin by letting $\Theta(t)$ model the actual phase of the carrier. This will include any frequency offset or phase jitter which is introduced by the channel. We may then let $\Phi(t)$ be the receiver's estimate of this phase. If we assume a noiseless channel and perfect equalization at the receiver, then the samples which are presented to the decision device, at time k, can be shown to equal [18]

$$q_k = e^{j(\Theta_k - \Phi_k)} a_k \tag{3.28}$$

where θ_k and Φ_k are samples of $\theta(t)$ and $\Phi(t)$, respectively. Consequently, when operating on this near perfect channel, the samples q_k will exactly equal their respective transmitted symbols when the estimate of the phase is equal to the actual phase. However, if the receiver demodulates the signal with a constant phase error, $\theta_k-\Phi_k=\Psi$, then all the received samples q_k will fall on a constellation that appears to be a tilted version of the original constellation. This is shown for a 4 phase shift keying (PSK) constellation in Figure 3.10. It causes a serious problem in both a coded and uncoded system because the rotation moves the received sample q_k away from its own signal point and closer to the incorrect neighbouring signal point. This will cause a reduction in the noise immunity of the system.

This problem is, however, greater for a TCM system. The greater severity can be seen by viewing the uncoded 4 PSK constellation and the expanded 8 PSK constellation shown in Figure 3.11. We see that since the signal points in the 8 PSK constellation are more closely spaced, an equal rotation of the received samples in both systems will bring the 8 PSK samples twice as close to their neighbouring incorrect symbol points as would be the case for 4 PSK samples.

This fact contributes to the bigger problem which can be seen by examining the transitions of a trellis diagram and by noting the operation of a Viterbi decoder. In the trellis, there exist long uninterconnected paths whose transitions are such that the Euclidean distance between them does not grow very fast. For instance, the paths may have many stages where their branches are associated with the same symbol (zero growth) or with symbols with the smallest distance Δ_0 between them. Without any phase error, the Euclidean distance between

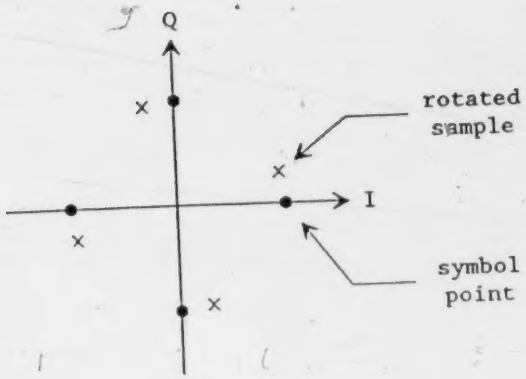


Figure 3.10 Signal constellation for the 4 PSK signal and the rotated constellation caused by a constant phase error

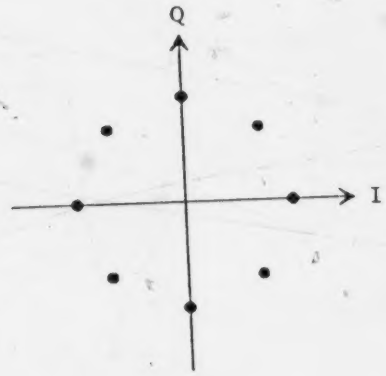


Figure 3.11 Signal constellation for the 8 PSK signal

the two paths would build up after a while and the Viterbi decoder would eliminate the incorrect path. If, however, a constant phase error occurs and causes the received samples to be located midway between the symbol points of the constellation then, by definition, the two distances from a received sample to its adjacent symbol points will be equal. These equal distances are simply the branch metrics. Consequently, branches which were different by a distance of Δ_0 with no phase error are now different by a distance of zero. This implies that the uninterconnected paths above may progress for a long time with no difference in their Euclidean distances. At this point, the code would be acting as if it were catastrophic and the decoder would begin to fail [30].

This phenomenon results in the performance of the coded 8 PSK signal deteriorating as the phase error, θ_k-Φ_k, increases. This occurs until the system completely fails at a phase error of 22.5°. In comparison, the uncoded 4 PSK system also suffers a performance reduction with increasing phase error but it does not completely fail until an error of 45° persists. This is one example showing the greater susceptibility of TCM systems to phase errors.

Another dramatic difference between the two systems is observed when there is indeed a failure. By examining the 4 PSK constellation we note that a rotation of 90, 180 or 270° results in the signal points realigning themselves. These phase ambiguities can crash the system if the transmitted symbols are not differentially encoded. For instance, normally a decision directed loop is used to track the carrier's phase. As a result, when the phase error is in the ±45° range, the tracking loop will try to drive the phase error towards 0°. If, however, the

phase error were to exceed one of these ±45° boundaries, the tracking loop would simply skip over and try to force the phase to equal 90 or 270°, respectively. If differential encoding is not used, the receiver will continue to operate normally but this flip will result in all of the symbols being received incorrectly. By differential encoding, only the relative phase of a received sample is important and hence, the number of errors caused by this phase skip will be small. When a system has this property, it is said to be phase or rotationally invariant, where the 4 PSK constellation is invariant to 90, 180 and 270° phase rotations.

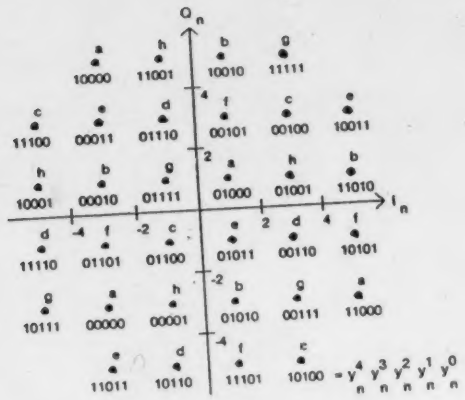
At first glance the coded 8 PSK constellation does not look much worse off. However, TCM codes are usually not invariant to all the phase rotations under which the constellation can be made phase invariant. In fact, the 4 state code using the 8 PSK constellation (identical convolutional encoder and trellis diagram as the 8 PAM signal) is only phase invariant to 180° rotations because of a lack of special symmetry (discussed shortly) in the trellis. This is in contrast to the constellation itself which can be made invariant to all phase rotations which are multiples of 45°. By the code only having one symmetry at 180°, any time the phase error is in the ranges between 22.5 and 157.5°, and 202.5 and 337.5°, the carrier-phase tracking loop will be in an undriven random-walk situation which can last for a long period of time [30]. This will result in a significant number of errors. Eventually, the system will resynchronize when the randomly fluctuating value of $\Phi(t)$ approaches a phase for which the received samples again resemble a valid TCM sequence. We should note that this can only happen around the two windows centred at 0 and 180°. Consequently, this

behaviour suggests that it would be desirable for TCM codes to be designed with as many phase symmetries as possible so that carrier-phase resynchronization after a temporary loss would be more rapid. These phase ambiguities, however, must not result in the wrong bits being decoded. Hence, some form of differential encoding and decoding should be applied to determine the proper quadrant of the symbols.

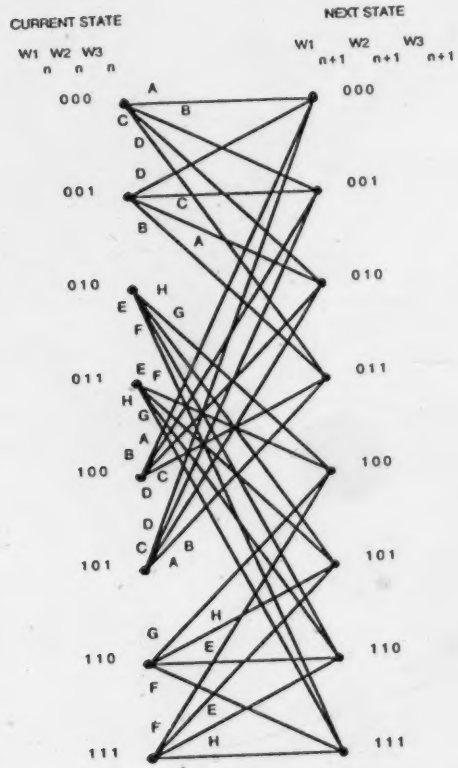
A breakthrough in designing TCM codes to be phase invariant was accomplished by L. F. Wei [31] [32]. He showed how non-linear elements in 8 state convolutional encoders could be used to produce codes with 90, 180 and 270° phase invariances without losing any coding gain.

Wei typically designed his codes in the reverse order of the steps taken by Ungerboeck. He would usually begin his design procedure by finding a trellis diagram with the properties he desired. For instance, he would find a trellis with the right number of states and with the transitions that would ensure a specific gain. These trellis diagrams would also have to satisfy a number of rules associated with the effects of phase rotations. His next step would be to find an appropriate signal constellation and map the signal points to the output bits of a convolutional encoder by again following a specific set of rules. Note, however, at this point the actual convolutional encoder was not yet known. The final step was to construct a truth table, using the current and next states of the trellis diagram, along with some of the input and output bits of the convolutional encoder, to actually generate the logic circuit diagram of the desired convolutional encoder.

An example of an eight state code with a gain of 4.0 dB is given in Figure 3.12. By first examining the signal's constellation, we can see the results of a number of Wei's rules. We begin by acknowledging



(a)



(b)

Figure 3.12 (a) 32 QAM constellation showing the bits to symbols mapping and symbol groupings for an 8 state code
(b) 32 QAM eight state trellis diagram

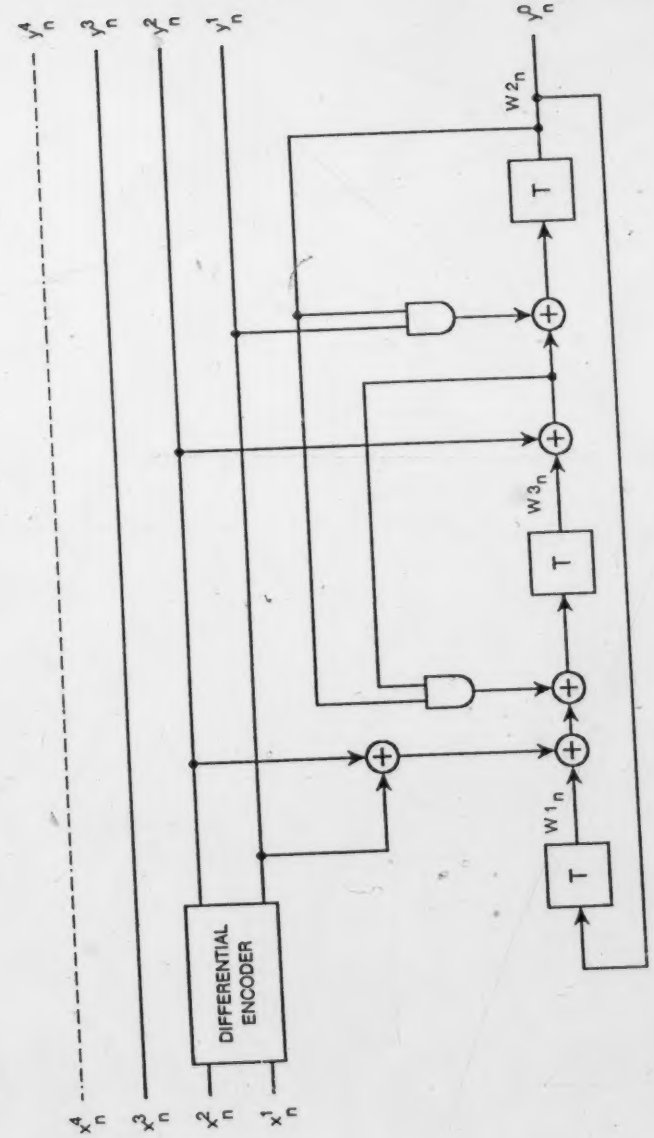


Figure 3.12 (c) 32 QAM eight state non-linear convolutional encoder

that there are eight subsets of symbols labeled a through g. All the symbols within a subset will have the same last three bits $y^2_n y^1_n y^0_n$. We next note, that by rotating the constellation clockwise in steps of 90°, the following subsets overlap a→e→c→g and b→f→d→h. If we now define eight new sets of symbols, where each new set contains four symbols which all have the same radius from the centre of the constellation but are 90° apart, we will observe two more of Wei's design rules. The first of which is, that the four symbols in each set must have the same values for $y^4_n y^3_n$. The second observation is that the bits $y^2_n y^1_n$, in each of these sets, differentially encode the quadrant. That is, these two bits change in the sequence 00→01→10→11 when moving in 90° steps clockwise.

We may now observe the trellis diagram in Figure 3.12b. What is important in this trellis, which is not obvious to see, is that any valid sequence of symbols A(k) can have its individual symbols rotated by 90, 180, or 270° and the resulting symbols will also be a valid TCM sequence. Note, however, that this sequence will not be the correct transmitted sequence of symbols. But because of the symmetry in the trellis and the symmetry in the constellation from following the previously given design rules, the decoded bits will be the correct bits which were transmitted. This is the key as to why these types of codes can be made transparent to 90, 180 and 270° phase rotations. Finally, by constructing a truth table with the current state variables and the differentially encoded bits on the input side and the next state variables and the encoder output y^0_n on the output side, the non-linear convolutional encoder in Figure 3.12c is produced. As a result of this design procedure, this TCM code will be phase invariant to 90, 180, and

270° rotations of its constellation and still produce a gain of 4.0 dB. However, it should be noted that the 8 state codes seem to be a special case. Similar non-linear codes with 16 states have been constructed but they have not been able to match the gain of the linear codes found by Ungerboeck.

Chapter 4

Computer Simulation Experiments on Digital Subscriber Loop Transmission

Now that we have the necessary background, we can discuss how the actual computer simulations were accomplished. This discussion will begin with an introduction to ABCD transmission matrices which were used to characterize uniform sections of digital subscriber loops. Following this, the generic transceiver structure used in both the baseband and passband simulations will be presented. Immediately after this, there will be an examination of the signal and noise models which are at the heart of this project. Finally, the two simulation outputs which are used to compare the performances of different systems, will be described.

4.1 Digital Subscriber Loop Model

4.1.1 Calculation of the Characteristic Impedance and Propagation Constant

As discussed in Chapter 2, digital subscriber loops are constructed out of uniform sections of 19, 22, 24 and 26 AWG cables. Each of these sections can be modelled with an ABCD transmission matrix (described in section 4.1.2). However, to calculate these matrices, we must have knowledge of the characteristic impedance Z_0 and the propagation constant γ for each section of the cable. These values can be found using the lumped-parameter model shown in Figure 4.1. This model becomes an exact model for a short section of transmission line as the length of the line dx approaches zero. By examining the figure, we

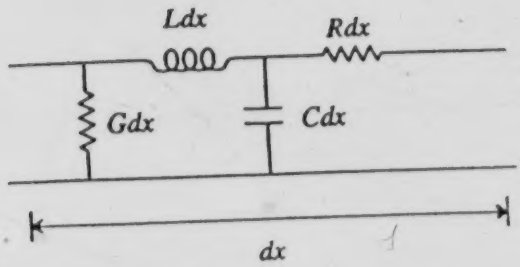


Figure 4.1 Lumped parameter model for a short section of transmission line

see that this model uses the four primary parameters R (resistance in ohms per unit length), L (inductance in henries per unit length), G (conductance in siemens per unit length) and C (capacitance in farads per unit length). In general, these parameters are all functions of frequency and are dependent on various factors such as the temperature, gauge and insulating material of the wire pair. Tables of these RLGC parameters, measured between the frequencies of 1 Hz and 5 MHz, for the 19, 22, 24 and 26 AWG polyethelene insulated cables (PICs) at 21° Celsius are given in Appendix B [33]. A larger set of similar tables for 22, 24 and 26 gauge PIC and pulp insulated cables at -18, 21 and 49° Celsius are available in [7].

From the tables in Appendix B, we see that of the four parameters only the capacitance is independent of frequency at 0.0516 μF/km. The resistance is found to be relatively constant at low frequencies but is proportional to the square root of frequency at higher frequencies because of the skin effect. The inductance is approximately 0.6 mH/km at low frequencies and decreases to about 80% of this initial value when the frequency increases to 1 MHz. Finally, the conductance is also seen to be frequency dependent but is very small and usually negligible for PIC cables.

From these primary parameters, the secondary parameters Z_0 (characteristic impedance) and γ (propagation constant) can be calculated using the following relationships

$$Z_0 = \left[\frac{R+j2\pi fL}{G+j2\pi fC} \right]^{0.5} \tag{4.1}$$

$$\gamma = [(R+j2\pi fL)(G+j2\pi fC)]^{0.5} \tag{4.2}$$

where f is the frequency. It should be noted that the values of γ are given per unit length. Z_0 and γ are explicitly dependent on frequency as shown above, but are also implicitly dependent because of the changes in the primary parameters with frequency.

A FORTRAN program called INTERPOL.FOR [34] was written to generate sets of Z_0 and γ values which are used in the modelling of subscriber loops. A single run of this program can produce many sets of these values where each set will be for a different combination of temperature and gauge of cable, but all of them will be for one particular baud rate. When the baud rate is changed, the program must be run again to produce additional sets of values because a change in the baud rate will cause a change in the frequency spacing between adjacent values. This will become more apparent in section 4.3.1 .

The way the program works can be seen by viewing the flow chart in Figure 4.2. Essentially, the program begins by reading a number NF, which is the number of sets of Z_0 and γ values it will have to produce. Following this, the program reads in the NF filenames containing the appropriate sets of R, L and G values to produce the desired sets of Z_0 and γ. The program will then read in the single value of the baud rate which will set the frequency spacing, Δf, between adjacent values of Z_0 and γ. The next major step is to enter the outer most loop in the program which will be re-entered for each of the NF sets of Z_0 and γ that must be produced. Once inside this loop, one array for each of R, L, G and FREQ, each containing N=38 non-uniformly spaced values, will be read by the program. The subroutine SPLINE [35] will then take each of these arrays separately and calculate their second derivatives at the frequencies given by FREQ. These values for the primary parameters'

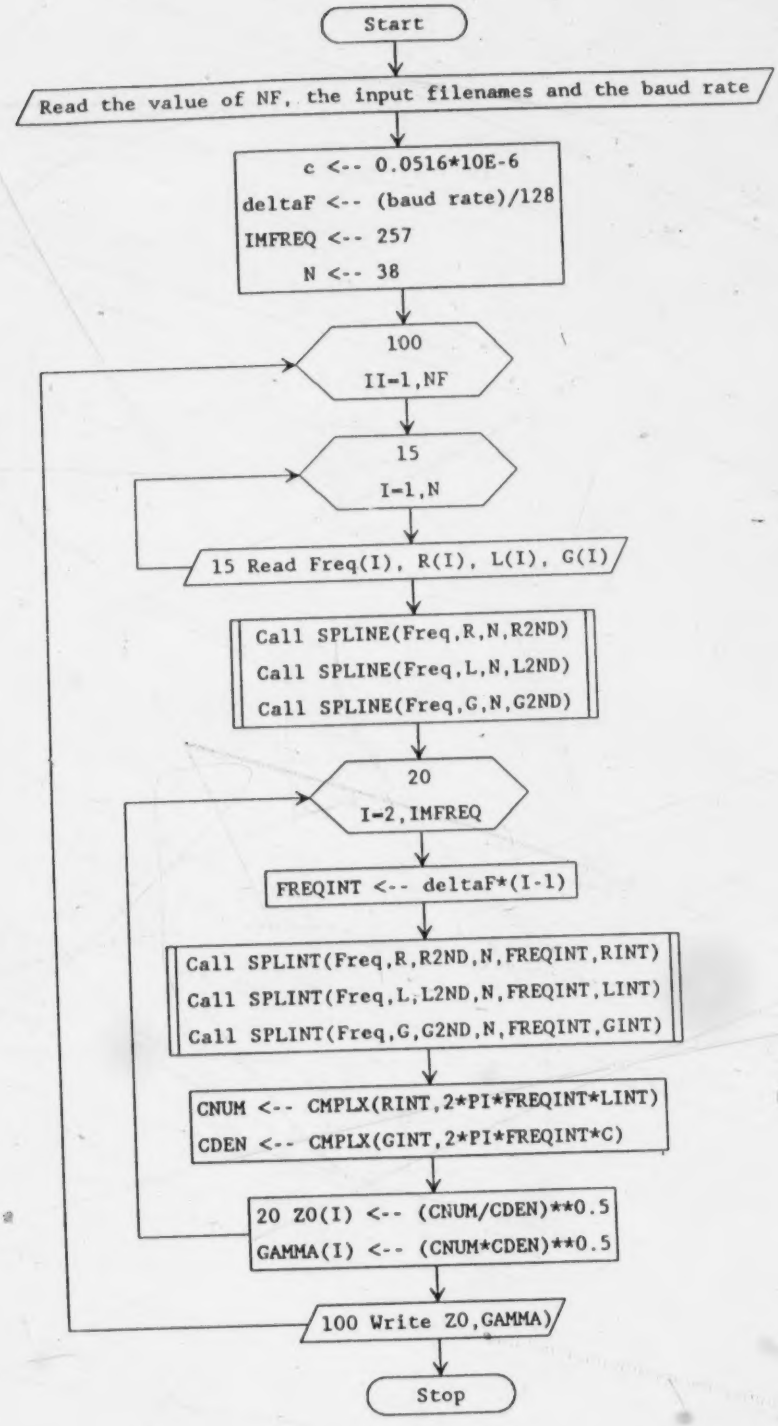


Figure 4.2 General flow chart for the program INTERPOL.FOR

second derivatives will be returned in the arrays R2ND, L2ND and G2ND, respectively. After this, the subroutine SPLINT [35] will use a cubic spline interpolation to find the intermediate values of R, L and G which are needed at frequencies which lie in between the tabulated frequencies. The interpolation algorithm uses a cubic polynomial given by

$$y(f)=Ay_i+By_{i+1}+Cy''_i+Dy''_{i+1} \tag{4.3}$$

where

$$A=\frac{f_{i+1}-f}{f_{i+1}-f_i} \qquad B=\frac{f-f_i}{f_{i+1}-f_i}$$

$$C=(A^3-A)(f_{i+1}-f_i)^2/6 \qquad D=(B^3-B)(f_{i+1}-f_i)^2/6$$

and y" denotes the second derivative. The values y_i and y_{i+1} at frequencies f_i and f_{i+1} would be the known or tabulated values which lie on either side of the desired interpolated value y at the frequency f. Once interpolated values for each of R, L and G are found at a necessary frequency, the program calculates the corresponding values of Z_0 and γ using Eqns.(4.1) and (4.2), respectively. When the program completes this inner loop, the values of Z_0 and γ for this gauge and temperature of cable will have been calculated for every frequency required. The program will then simply write these values to a file and branch back to its beginning where it will read in a new set of R, L and G values which will be used to calculate another set of Z_0 and γ values.

4.1.2 Calculation of ABCD transmission matrices

As stated in the previous subsection, digital subscriber loops are composed of several uniform sections of cable. Each of these single

gauge sections can be thought of as a two port network as shown in Figure 4.3. An ABCD matrix is used to relate the input voltage and current to the output voltage and current through the relationships

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \tag{4.4}$$

In general, all of these quantities are complex functions of frequency. It can be shown that the elements of an ABCD matrix, for a uniform section of cable, are given by [18]

$$A = \cosh(\gamma L) \qquad B = Z_0 \sinh(\gamma L)$$
$$C = \frac{\sinh(\gamma L)}{Z_0} \qquad D = \cosh(\gamma L) \tag{4.5}$$

where L is the length of the uniform section and Z_0 and γ are as defined in the previous subsection.

The importance of ABCD matrices stems from the fact that when two twoport networks are connected in series, the overall relationship between the input voltage and current and the output voltage and current can be obtained by finding the product of the first network's ABCD matrix times the second network's ABCD matrix (order of the multiplication is very important). As a result of this fact, we can take any number of single gauge sections of cable, that are connected in series, and model them with a single composite ABCD matrix. This can be expressed analytically as

$$\begin{bmatrix} A_{comp} & B_{comp} \\ C_{comp} & D_{comp} \end{bmatrix} = \prod_{i=1}^{N} \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \tag{4.6}$$

where there are a total of N sections. This description is perfect for a digital subscriber loop with no bridged taps. However, as we already

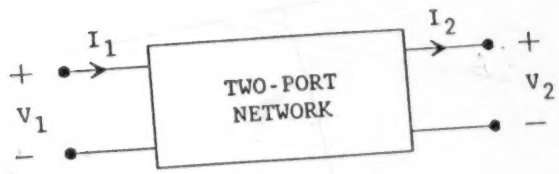


Figure 4.3 Illustration of a two-port network with the definitions of its voltages and currents

know, most subscriber loops have at least one bridged tap which is a section of cable which is in parallel with the main cable pair. This obviously does not fit into the requirement that the section be connected in series. The solution to this problem is to transform the parallel section into a series section. The way this is accomplished is described in Appendix C, while the result which is the equivalent series ABCD matrix for a bridged tap is given by

$$\begin{bmatrix} A_{sbr} & B_{sbr} \\ C_{sbr} & D_{sbr} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C_{compbr}/A_{compbr} & 1 \end{bmatrix} \tag{4.7}$$

where A_{compbr} and C_{compbr} are the composite-A and C values for the entire bridged tap. In the simplest case, where the bridged tap is a single uniform section of cable, A_{compbr} will equal $\cosh(\gamma_b L_b)$ and C_{compbr} will equal $\sinh(\gamma_b L_b)/Z_{0b}$ where L_b is the length of the bridged tap, and Z_{0b} and γ_b are the characteristic impedance and propagation constant of the bridged tap, respectively. Once the equivalent series matrix is calculated, it can be inserted into the main cable pair and be treated as just another matrix by Eqn.(4.6).

A FORTRAN program called DTXLINE.FOR [34] was written to calculate the composite ABCD matrices for all the digital subscriber loops simulated. The only limitation of the program is that it only allows two levels of bridged taps on a subscriber loop. That is, it will not simulate a loop that has a bridged tap on a bridged tap on a bridged tap as shown in Figure 4.4. The program could easily be modified to simulate such a loop except these types of loops are rare and are of no special interest.

As inputs, the program requires the configuration of the loop and the values of Z_0 and γ for each section of cable used to construct the

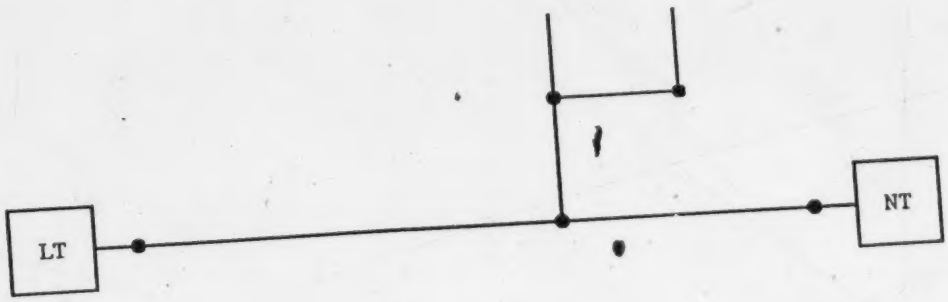
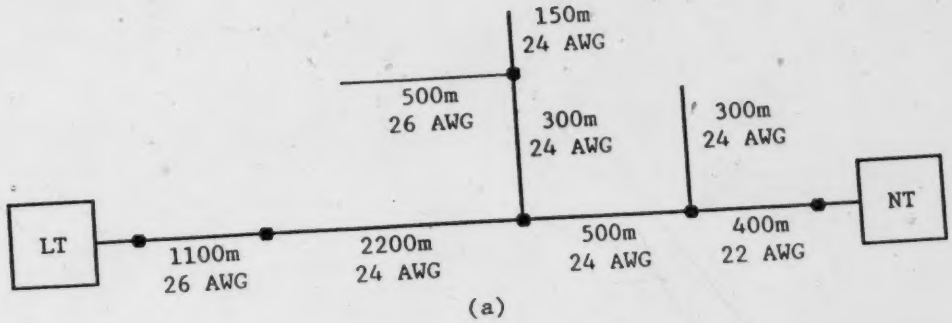


Figure 4.4 An example of a loop with three levels of bridged taps

loop. The equations from which the values of Z_0 and γ can be obtained have already been discussed. Consequently, we will now concentrate on how the configuration of the loop is communicated to the program. Figure 4.5 shows a subscriber loop and the input file that would be read by the program. We see that the input file has a total of 4 columns and a number of rows. The first column of each row is for a control digit which can take on the values of 0, 1 or 2. If the control digit is equal to 0 or 2, then the row is only a control statement and the other columns are meaningless. If the digit is a 1, then the row will represent a section of cable and its remaining three columns will give the values for the cable's gauge, temperature and length, respectively. The meaning of the control digit's value in different situations and the corresponding sequences of actions the program will follow can be seen by viewing the general flow chart in Figure 4.6. A summary of this flowchart is as follows.

When the program is on the main line and reads in a row with a control digit equal to '0', the program knows that the loop has ended and the composite ABCD matrix for the loop should be written to a file. If the digit is equal to '1', then this row represents a cable section which should be placed in series with the rest of the main line. This is accomplished by calculating the ABCD matrix of this newest section and multiplying it by the composite ABCD matrix of the loop up to this point. If the control digit read by the program is a '2', then the program knows that there is a bridged tap at this point on the main loop.

When the program is on a bridged tap and reads in a row with a control digit equal to '0', then it is known by the program that the



(a)

1	26	21	1100.0
1	24	21	2200.0
2	0	0	0.0
1	24	21	300.0
2	0	0	0.0
1	26	21	500.0
0	0	0	0.0
1	24	21	150.0
0	0	0	0.0
1	24	21	500.0
2	0	0	0.0
1	24	21	300.0
0	0	0	0.0
1	22	21	400.0
0	0	0	0.0

(b)

Figure 4.5 (a) Illustration of a subscriber loop's structure
(b) Corresponding input file to the program

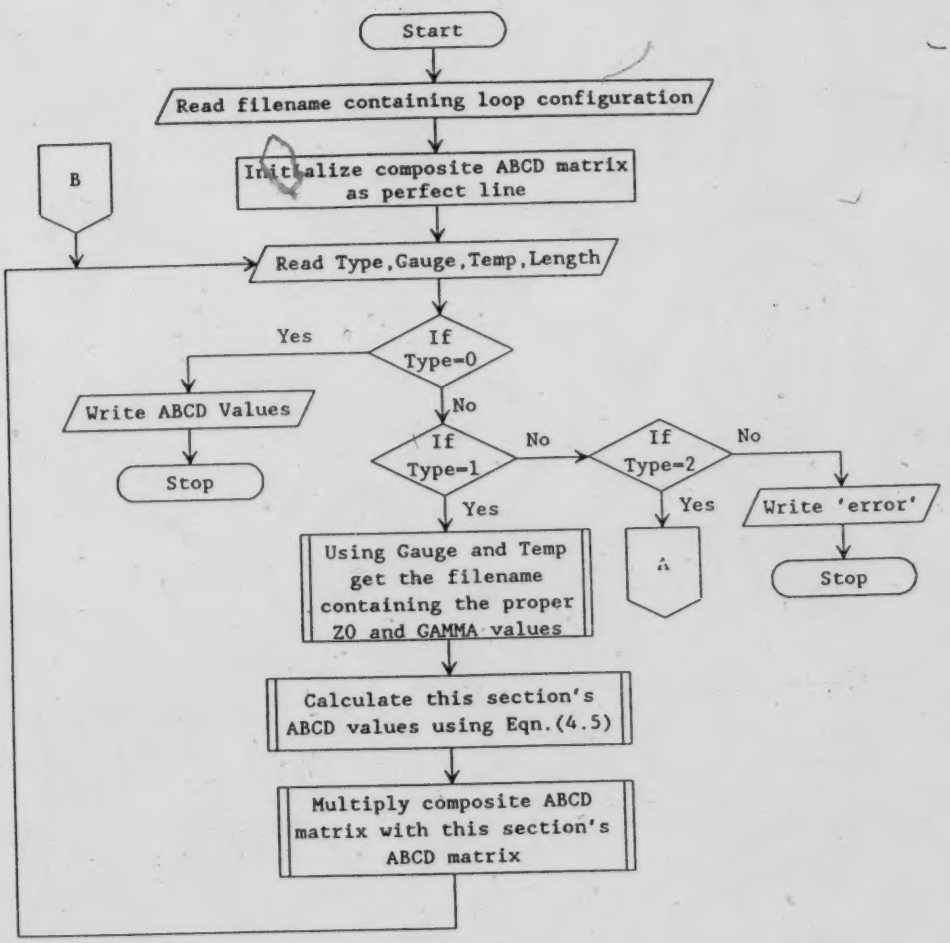


Figure 4.6 General flow chart for the program DTXLINE.FOR

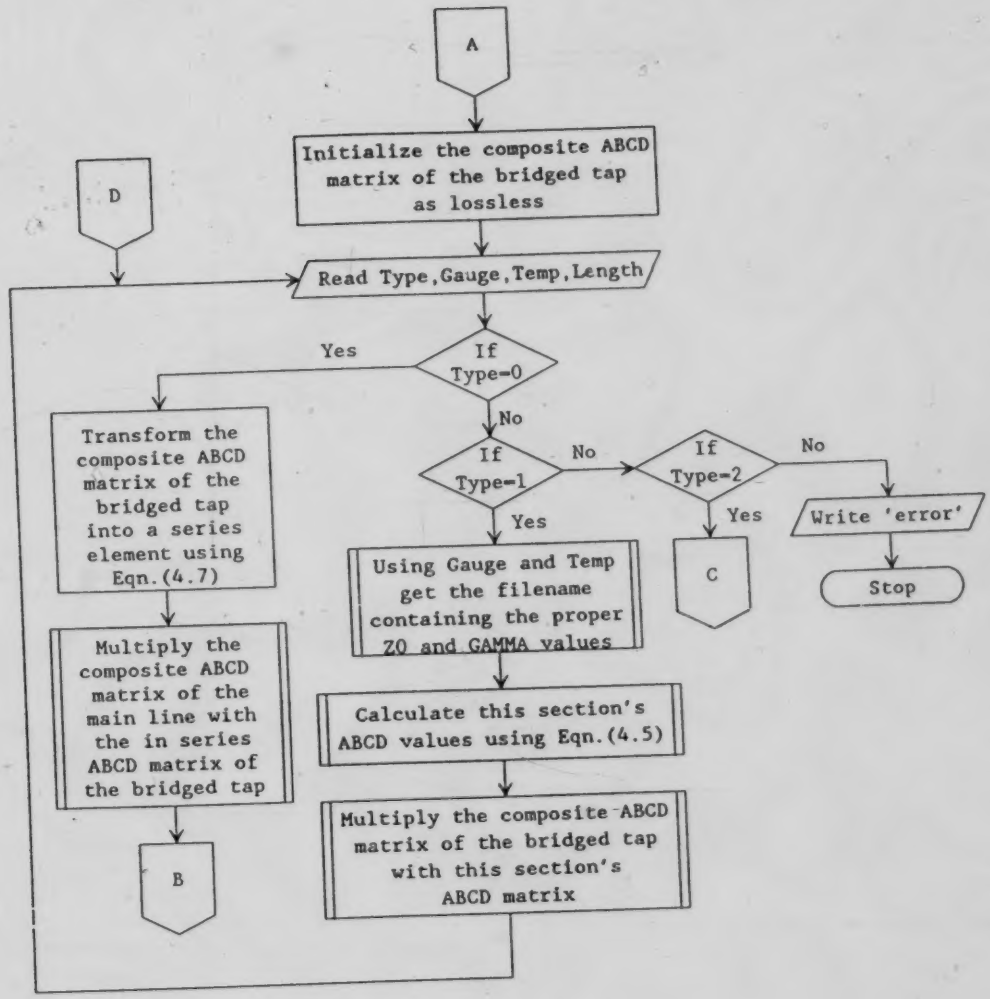

A
Initialize the composite ABCD matrix of the bridged tap as lossless
D
Read Type,Gauge,Temp,Length
If Type=0
Yes
No
Transform the composite ABCD matrix of the bridged tap into a series element using Eqn.(4.7)
Multiply the composite ABCD matrix of the main line with the in series ABCD matrix of the bridged tap
B
If Type=1
No
Yes
If Type=2
No
Yes
Write 'error'
Stop
C
Using Gauge and Temp get the filename containing the proper Z0 and GAMMA values
Calculate this section's ABCD values using Eqn.(4.5)
Multiply the composite ABCD matrix of the bridged tap with this section's ABCD matrix

Figure 4.6 General flow chart for the program DTXLINE.FOR (Con't)

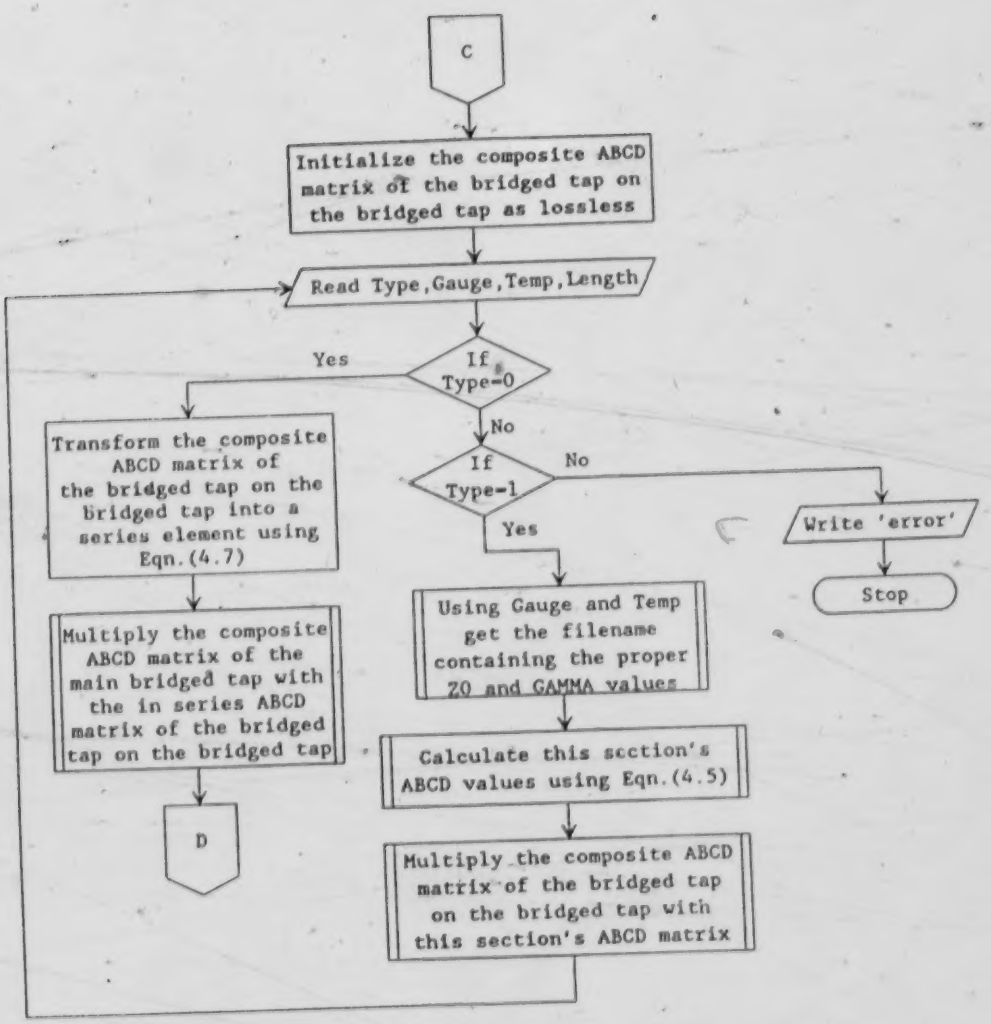

C
Initialize the composite ABCD matrix of the bridged tap on the bridged tap as lossless
Read Type,Gauge,Temp,Length
If Type=0
Yes
No
Transform the composite ABCD matrix of the bridged tap on the bridged tap into a series element using Eqn.(4.7)
Multiply the composite ABCD matrix of the main bridged tap with the in series ABCD matrix of the bridged tap on the bridged tap
D
If Type=1
No
Yes
Write 'error'
Stop
Using Gauge and Temp get the filename containing the proper Z0 and GAMMA values
Calculate this section's ABCD values using Eqn.(4.5)
Multiply the composite ABCD matrix of the bridged tap on the bridged tap with this section's ABCD matrix

Figure 4.6 General flow chart for the program DTXLINE.FOR (Con't)

bridged tap has ended and its composite ABCD matrix should be transformed to a series matrix and multiplied by the composite ABCD matrix of the main loop up to this point. If the digit is equal to '1', then this row represents a section of cable which should be placed in series with the rest of the bridged tap. Finally, if the digit is a '2', then there is a bridged tap from the bridged tap.

When the program is on a bridged tap from a bridged tap, a control digit equal to '0' tells the program that this second level bridged tap has ended and its composite ABCD matrix should be transformed into a series matrix and multiplied by the composite ABCD matrix of the first level bridged tap. If the control digit is a '1' then this row again represents a section of cable, but now it must be placed in series with the rest of the second level bridged tap. At this time, we again note that a third level of bridged taps is not allowed and thus a control digit equal to '2' at this point is not allowed.

We should note that since the order of the ABCD matrix multiplications is important, the final composite ABCD matrix produced by this program will be for one direction of transmission (line termination to network termination). Appendix D shows that the ABCD matrix for the other direction of transmission (NT to LT) can be obtained by simply interchanging the A_{comp} and D_{comp} terms of this composite ABCD matrix produced by the program.

4.2 Transceiver Structure

A simplified generic block diagram of the full-duplex digital subscriber loop transceiver used in the simulations is given in Figure 4.7. We may begin its description by examining the blocks in the

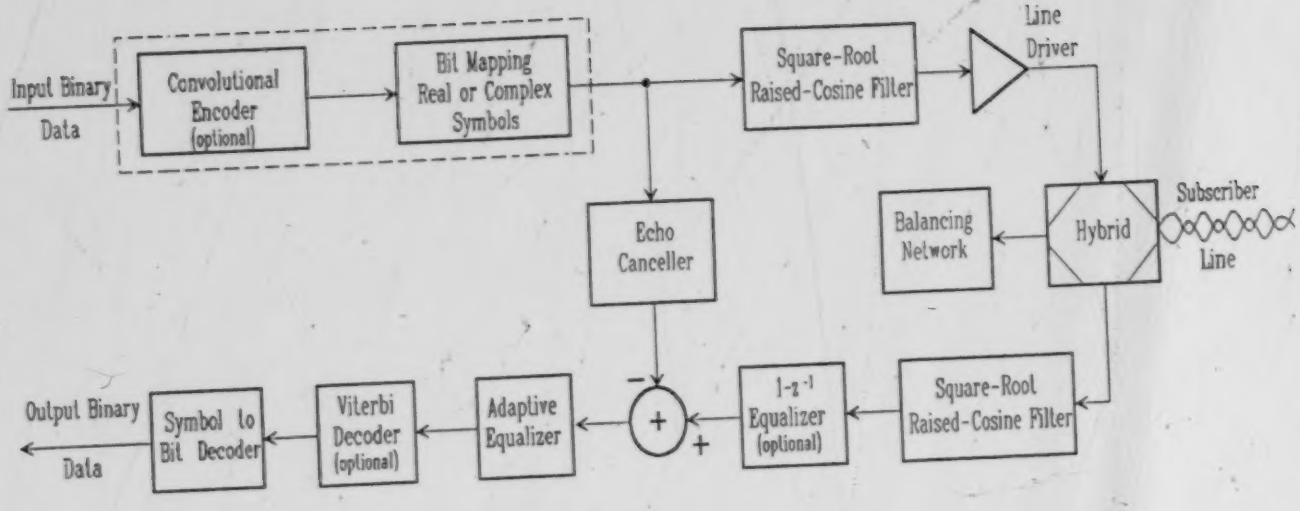


Figure 4.7 Generic block diagram of the transceiver assumed in the simulation

transmission path which is shown here as the upper path in the figure. When baseband or passband trellis coded modulation was used, the input bits were convolutionally encoded before they were sent to the bit to symbol mapper. This introduced some redundancy into the system, as was described in section 3.3.1. In systems where there was no trellis encoding, the convolutional encoder was skipped and the input bits went directly to the mapper. In either case, the input bits were mapped to real-valued symbols in a baseband system and were mapped to complex-valued symbols in a passband system. Once again, as this implies, the passband signals were modelled as equivalent complex-valued baseband signals as was described in section 3.2. The various bit to symbol mappings for the coded and uncoded signals which were simulated are shown in Figures 4.8 through 4.17. In the cases where the signal constellations are expanded (trellis coded), the associated convolutional encoders and trellis diagrams are also shown. These TCM signals were designed following the rules discussed in sections 3.3.1 and 3.3.5.

The first baseband transceiver we considered used PAM and the 2B1Q line code without trellis encoding. Its constellation is shown in Figure 4.8. This is the line code which was specified in the ISDN basic access interface standards [7]. Consequently, this transceiver is a good reference from which we could measure the gain (or loss) in performance by making the other transceivers more complicated. By expanding this constellation we obtained the 8 PAM signal set which is shown with its symbol groupings in Figure 4.9. The convolutional encoder and trellis diagram for the 4 state code is shown in Figure 4.10 while those for the 8 state code are shown in Figure 4.11. These codes

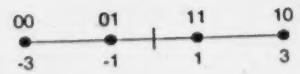


Figure 4.8 Uncoded 4 PAM constellation

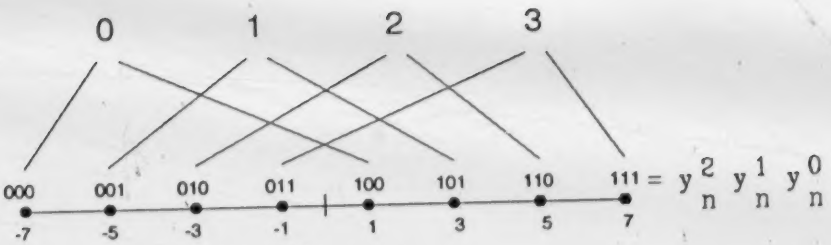
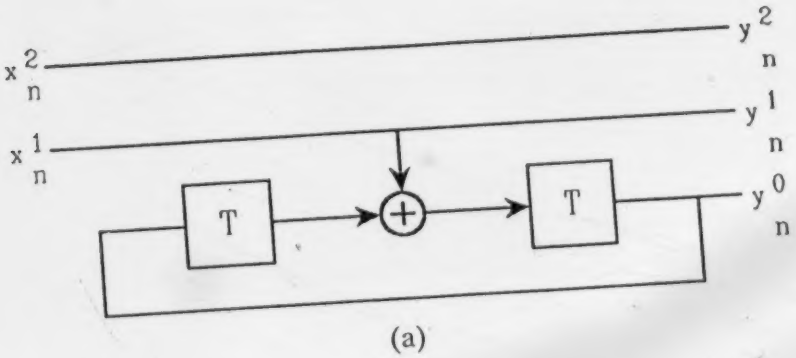
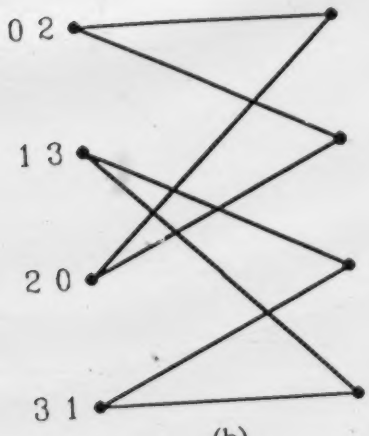


Figure 4.9 Coded 8 PAM constellation with bit mapping and symbol groupings



(a)



(b)

Figure 4.10 (a) 8 PAM four state convolutional encoder
(b) 8 PAM four state trellis diagram

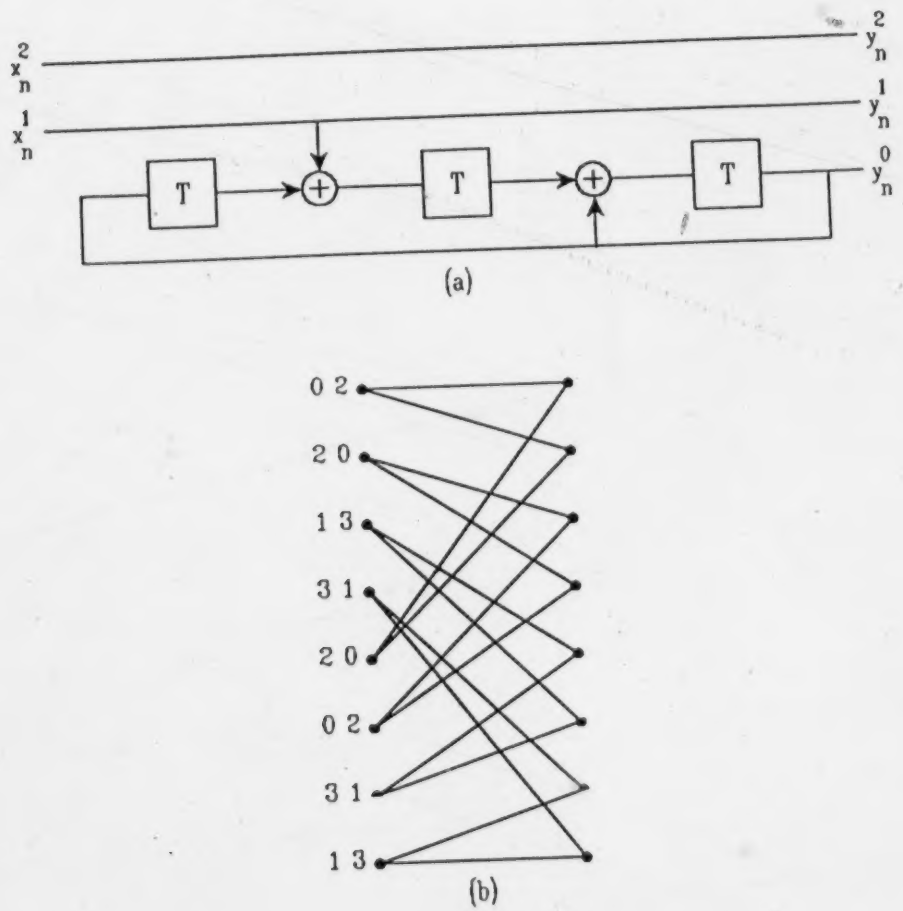


Figure 4.11 (a) 8 PAM eight state convolutional encoder
(b) 8 PAM eight state trellis diagram

have an asymptotic gain of 3.31 dB and 3.77 dB over an uncoded system, respectively.

The next transceivers modelled were passband transceivers. The first of the two uncoded systems simulated used the 8 QAM constellation shown in Figure 4.12. This signal has an ideal spectral efficiency of 3 bits/sec/Hz. When trellis encoding was applied, the resulting 4 and 8 state 16 QAM constellations with their respective bit to symbol mappings are shown in Figure 4.13. The convolutional encoders and trellis diagrams associated with these two constellations are shown in Figures 4.14 and 4.15, respectively. The 4 state code provides a gain of 3.0 dB over an uncoded system and is phase invariant to 180° rotations of its constellation. The 8 state code has a gain of 4.0 dB and is invariant to 90, 180 and 270° rotations.

The second uncoded passband system simulated used a 16 QAM signal with the constellation shown in Figure 4.16. It has an ideal spectral efficiency of 4 bits/sec/Hz. When trellis encoding was applied to this signal set, the resulting 4 and 8 state 32 QAM constellations are shown in Figure 4.17. These codes have the same convolutional encoders and trellis diagrams as those shown in Figures 4.14 and 4.15, except there is one additional bit which passes through the convolutional encoder uncoded. The 4 state code is again invariant to 180° phase rotations while the 8 state code is again invariant to 90, 180 and 270° rotations. These codes have asymptotic coding gain of 3.0 and 4.0 dB, respectively.

The next block in Figure 4.7 shows a square-root raised cosine filter. This filter has a number of functions. In the time domain, it determines the shape of a basic pulse as was discussed in section 2.1.2. In the frequency domain, it bandlimits the signal to reduce the effects

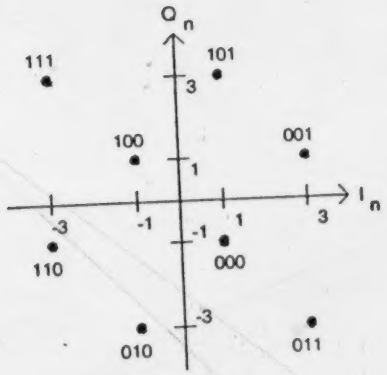
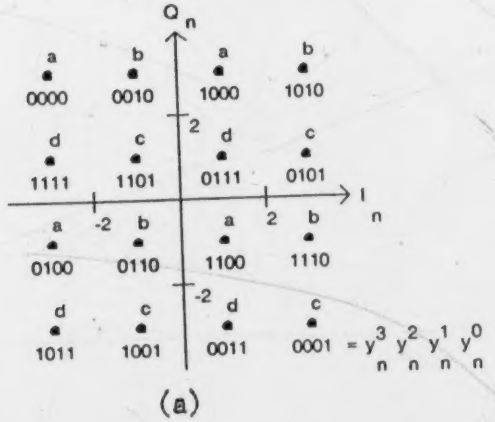
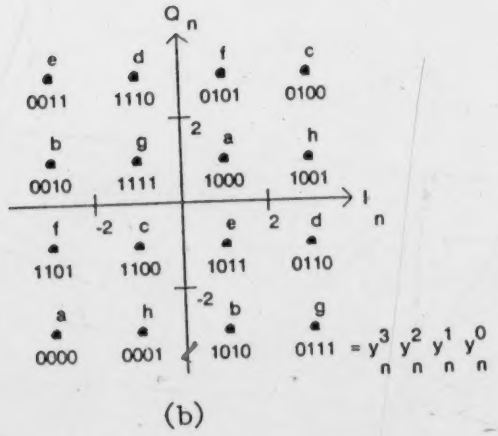


Figure 4.12 Uncoded 8 QAM (AMPM) constellation

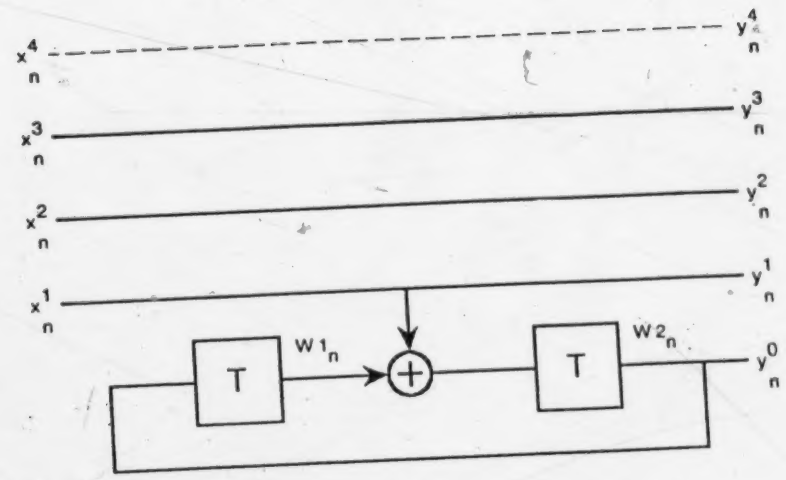


(a)

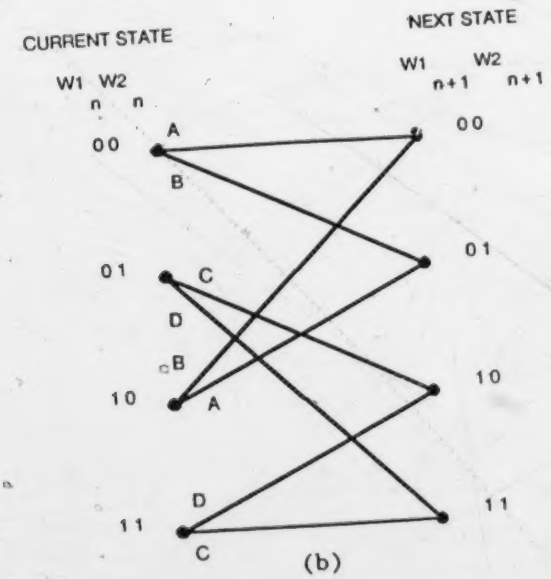


(b)

Figure 4.13 (a) Coded 16 QAM four state constellation
(b) Coded 16 QAM eight state constellation

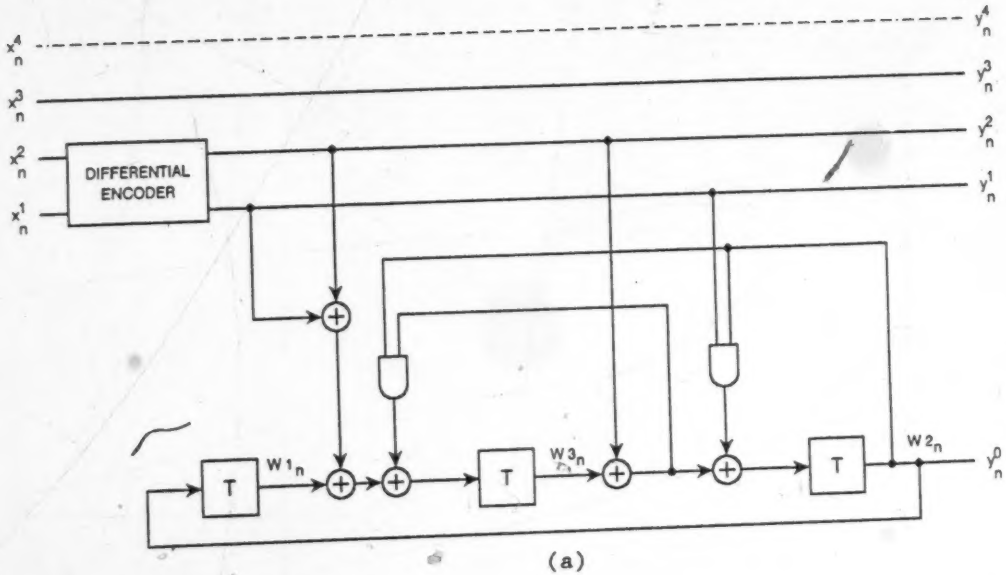


(a)



(b)

Figure 4.14 (a) 16 QAM and 32 QAM four state convolutional encoder
(b) 16 QAM and 32 QAM four state trellis diagram



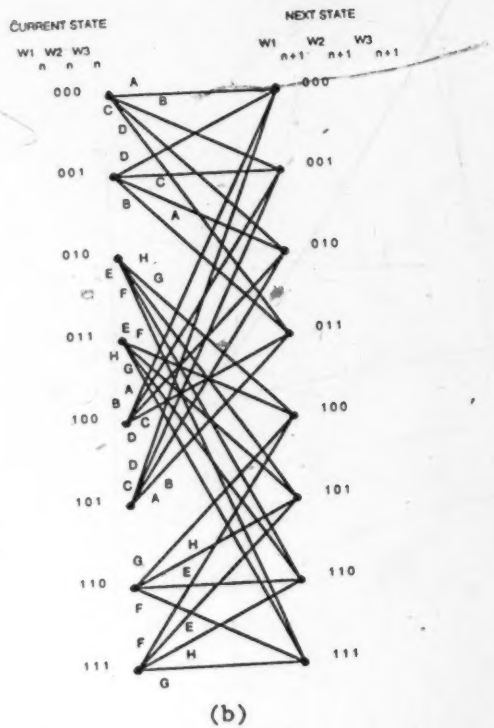


Figure 4.15 (a) 16 QAM and 32 QAM eight state convolutional encoder
(b) 16 QAM and 32 QAM eight state trellis diagram

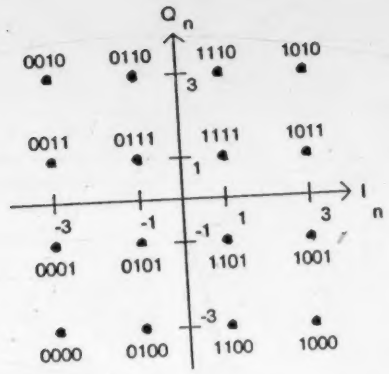
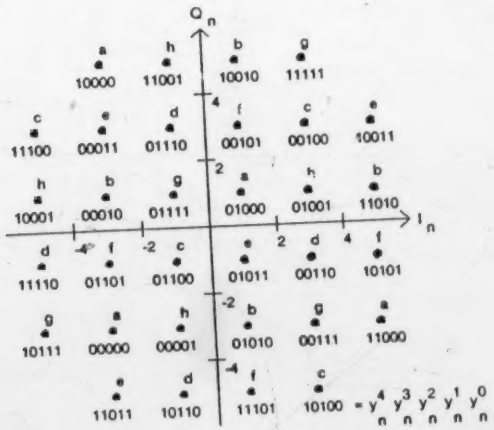
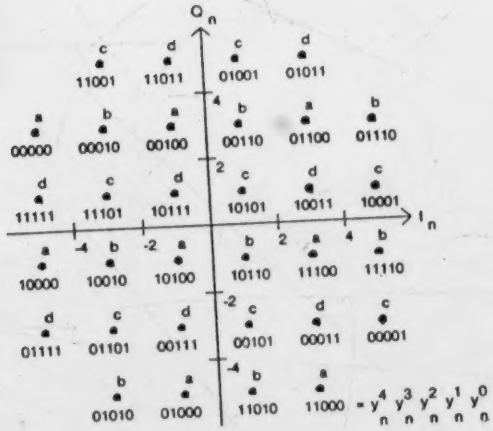


Figure 4.16 Uncoded 16 QAM constellation



(a)



(b)

Figure 4.17 (a) Coded 32 QAM four state constellation
(b) Coded 32 QAM eight state constellation

of crosstalk and it also shapes the power spectrum of the transmitted signal.

The line driver, which is shown next, amplifies the transmitted signal so that the average power at both the line and network interfaces is at a level of 13.5 dBm over a 135 ohms test resistance. This value is part of the basic access interface standard and was used in all the simulations to ensure that any performance comparisons between different systems would be done fairly.

The final block in the transmission path is the hybrid circuit. As was discussed in section 2.2.1, its functions are to isolate the transmitted and received signals and perform a two wire to four wire conversion. The actual balancing network used, within the hybrid circuit, is the 5 element network shown in Figure 4.18. This network typically provided 10 to 15 dB of transhybrid attenuation of the local echo for most loops. The additional echo suppression required by a system was provided by an echo canceller. In these simulations, the function of an echo canceller was modelled as a direct reduction of the echo level by an amount considered reasonable for a state-of-the-art echo canceller.

The first block in the receiver path is another square-root raised cosine filter. This filter removes any out of band noise and prevents aliasing when the continuous received signal is later sampled. In addition to this, by placing a square-root raised cosine filter in both the transmitting and receiving paths, the two filters have an overall transfer function which satisfies Nyquist's first criterion for no ISI. This does not imply that the two filters will remove all the ISI but rather they can help to reduce it. In these simulations, all of the

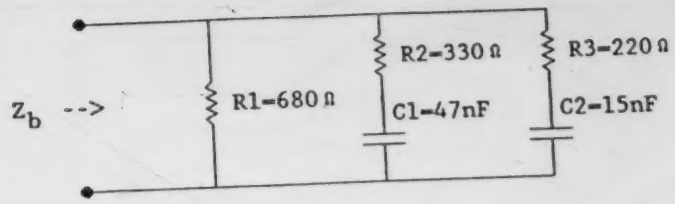


Figure 4.18 Balancing network used within the hybrid circuit

baseband transceivers used filters with a 100% roll-off factor and all of the passband transceivers used filters with a 20% roll-off factor. These values were chosen after comparing the relative performances of a number of systems which used different roll-off factors. The systems which used these particular values were found to have superior performances.

The next block in the receiver path is a multiple response fixed pre-equalizer. It has the transfer function $P_e(z)=1-z^{-1}$ and was introduced into only the baseband transceivers. The function of the pre-equalizer is to reduce the length of the tail of the system's impulse response. Its effectiveness can be seen in Figure 4.19 where the pulse responses of a 2100m section of 26 AWG cable with and without pre-equalization are shown. The pre-equalized pulse has a shorter and smaller amplitude tail than of the pulse without pre-equalization. This shorter tail results in the echo canceller and the decision feedback equalizer being of reduced complexity by virtue of having fewer taps. It should be noted that the passband transceivers did not use this pre-equalizer because their impulse responses were relatively short compared to the baseband impulse responses.

After this, some form of adaptive equalization was used on both the baseband and the passband signals. Two different equalizer structures were used with all the baseband transceivers. The first structure simply used an adaptive DFE with only the fixed pre-equalizer as a forward equalizer. The second structure used both the fixed pre-equalizer and an adaptive T/2 fractionally spaced linear equalizer (FSLE) prior to the DFE. The DFE section in both equalizers had 16

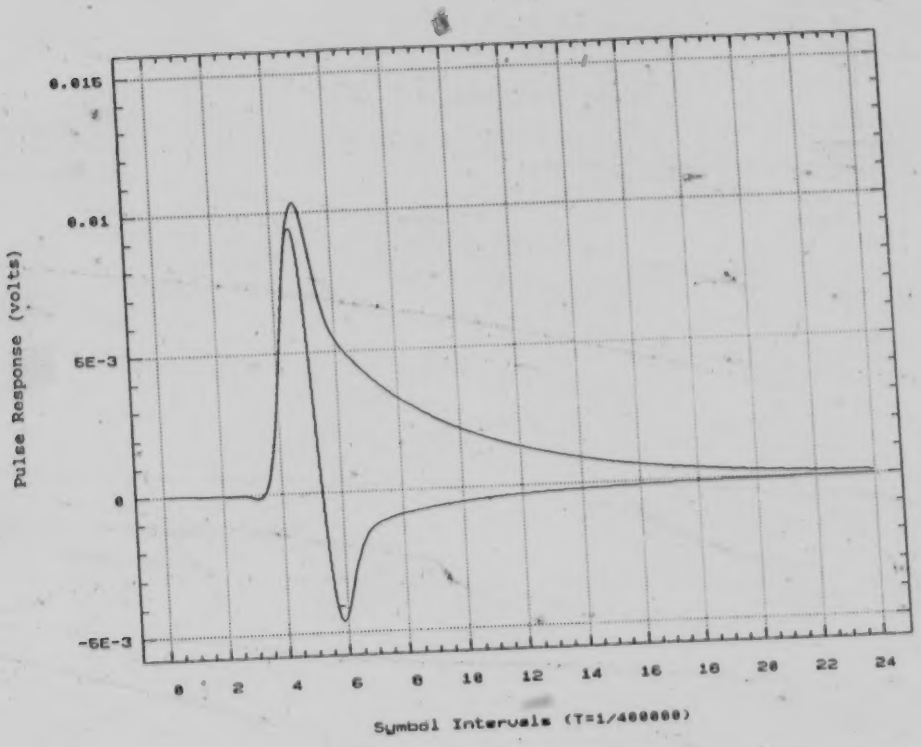


Figure 4.19 Pulse responses of a 2100m 26 AWG cable with and without a pre-equalizer

taps, while the fractionally spaced forward equalizer had 21 taps. These numbers of taps were found to produce the best results.

Three different equalizers were simulated with the passband transceivers. The first was a symbol spaced linear equalizer, with a total of 21 taps. The second was a T/2 FSLE which also had 21 taps, and the third equalizer simulated had a 21 tap forward T/2 FSLE with a 10 tap DFE. It should also be mentioned that some simulations were carried out using a fourth equalizer, an ISI canceller. Its structure was fairly complex. Its first equalizer consisted of a 21 tap forward T/2 FSLE with a 10 tap DFE. This equalizer's preliminary decisions were sent to a second equalizer which consisted of another 21 tap forward T/2 FSLE with both a 10 tap DFE and a 10 tap precursor cancelling filter as was described in section 2.1.3.2.2. Although it was considerably more complex, the ISI canceller only had a marginally better performance than third equalizer simulated. In other words, the whole structure did not perform much better than just its first equalizer which made the preliminary decisions. Consequently, no further simulations were undertaken with this equalizer structure. Finally, it should be noted that since the passband signals were modelled as equivalent complex-valued baseband signals, these equalizers had complex-valued taps.

Following the equalizers came a Viterbi decoder but only when TCM was used. Its operation was described in section 3.3.2. It should be recalled that when the equalizer has a DFE section then the decoding and equalization are done together using the parallel decision feedback decoder described in section 3.3.3.

The final block to be described in the transceiver is the symbol to bit decoder. Depending on whether TCM is used or not, this decoder

will take symbol decisions from either the Viterbi decoder or from a decision making device at the output of the adaptive equalizer. Its function is to decode the received symbols into bits. This will give the final binary output shown at the end of the receiver path.

As a final comment to the simplified transceiver structure in Figure 4.7, we note that there are no blocks showing either carrier or timing recovery circuits. These circuits were assumed to be functioning ideally in the simulations. This can be justified by considering that the purpose of this thesis is to determine, at the system's level, which modulation techniques (coded and uncoded) are best suited for the digital subscriber loop environment. Although the recovery circuits are very important, at this point in the research they are of secondary significance.

4.3 Transceiver Model

By using both the transceiver structure, just described, and the knowledge that any subscriber loop can be modelled with a composite ABCD matrix, we can derive the transfer functions of the signals and noises used in these simulations. From each of these transfer functions we can obtain respective impulse responses which can then be convolved with an appropriate input sequence to generate either a received signal or one of a number of corrupting additive noises. This can be seen in Figure 4.20 where the simulated signal processing structure for one direction of transmission is shown. We see that there are a total of four distinct paths which are eventually summed together. The sequences of blocks in each of the paths illustrate the routes which the received signal, echo, coloured Gaussian noise and near-end crosstalk (NEXT) must

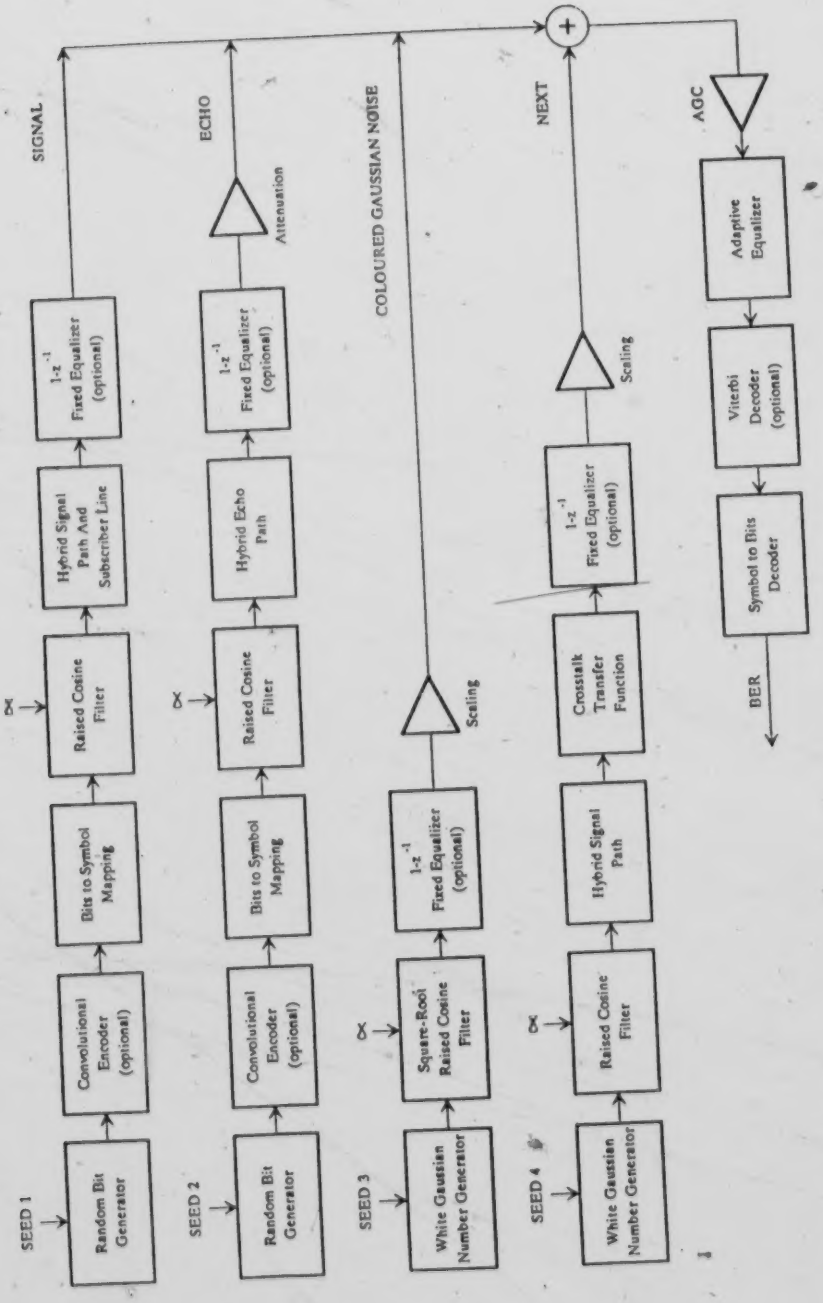


Figure 4.20 Simulated signal processing structure for one direction of transmission

take to reach the adaptive equalizer in the receiver. Once equalization is performed the sampled symbols are decoded to produce a bit stream, from which a bit-error-rate (BER) can be determined. The details of all these processes will be discussed in the following subsections.

4.3.1 Received Signal Model

By recalling the generic transceiver structure of Figure 4.7, we note that a signal which is transmitted from one end of a subscriber loop to the other, must pass through the square-root raised cosine transmit filter, the line driver, a hybrid, the subscriber loop, another hybrid, the square-root raised cosine receive filter and, if it is a baseband system, it must also pass through the multiple response fixed pre-equalizer. Since all of these devices are assumed to be linear, they can be lumped together as one equivalent discrete time channel. This is shown in Appendix F where the composite transfer functions of the received signal paths (one for each direction of transmission) are obtained by multiplying together the individual transfer functions of these devices. Once a signal path transfer function is known, we can promptly find its impulse response. In the simulation, the first step in accomplishing this goal is to take the fast Fourier transform (FFT) of a 2048 point data array containing two opposite polarity impulses. The 2048 samples come as a result of 128 symbols being sampled at 16 samples per symbol. The two impulses are separated by the maximum distance possible of 1024 samples (64 symbols) to ensure that their resulting responses do not overlap. The decision to use both a positive and negative impulse was made so as to ensure that the final signal path impulse response would have no d.c. offset. The next step in finding

this signal path impulse response is to multiply the complex spectrum of the two impulses by the transfer function of the received signal path. Note, however, that both of these multiplicands are generated at discrete frequencies and care must be taken to ensure that the frequency components which are multiplied together are for the same frequency. The frequency spacing produced by the FFT and consequently the spacing required by the transfer function is given by

$$\Delta f = \frac{1}{NS \cdot T} \tag{4.8}$$

where 1/T is the baud rate and NS is the number of symbols which are sampled at SS samples per symbol to produce the NN=NS·SS=2048 point data array. Although the value of NS is the same in all the simulations, the frequency spacing Δf will change when the symbol rate 1/T changes. This is the reason why the frequency spacing between the RLGC parametres which produce the ABCD matrices which contribute to the signal path transfer function is so important.

Once the frequency domain multiplication is done, an inverse Fast Fourier Transform (IFFT) is performed on the product to produce a received signal path impulse response which is contained in the first 1024 points of the data array. This response will be truncated and eventually used as a finite impulse response (FIR) filter. In the simulations using baseband transceivers, the responses were truncated to a length of 33 symbol periods. The responses for the passband transceivers were truncated to a length of 17 symbol periods. These lengths were chosen after examining many impulse responses where it was found that the tails of the responses were negligible beyond these points.

This description for finding the signal path impulse response, essentially applies to both the baseband and passband simulations. However, before we continue, there are a few differences that must be acknowledged. First of all, since the passband signals are being modelled as equivalent complex-valued baseband signals, the transfer function of the signal path must be translated down in frequency before the multiplication. That is, the frequency components centred at the carrier frequency must be shifted down so that they are centred at d.c. In these simulations, the carrier frequency was always chosen to equal $(1+\alpha)/2T = 0.6/T$. Hence, the frequency components of the signal path must be shifted down by $0.6/T$ as is described in section 3.2.

The second difference between the baseband and passband simulations comes as a result of having to demodulate the passband signals. As previously stated, a QAM signal requires a coherent demodulator. This means that the demodulating signal should have the exact carrier frequency and phase as the received signal. In the passband simulations, we know precisely what the carrier frequency is at the transmitter and hence, we can simply use it at the receiver. The phase shift, however, will depend on the delay introduced by a particular subscriber loop and can have any value between $-\pi/2$ and $3\pi/2$ radians. Since this thesis is not concerned about the design of a good phase tracking circuit, we can find the phase shift introduced by a particular loop by examining its transfer function. By calculating the phase of the transfer function's frequency component at the carrier frequency, we will have obtained the phase shift a sinusoid at this frequency would experience by passing over this loop. Consequently, the expression for the demodulating signal's phase is given by

$$\theta_{del} = \begin{cases} TAN^{-1}\{Im(H_s(f_c))/Re(H_s(f_c))\} & \text{if } Re(H_s(f_c))>0 \\ \pi/2 & \text{if } Re(H_s(f_c))=0 \\ \pi+TAN^{-1}\{Im(H_s(f_c))/Re(H_s(f_c))\} & \text{if } Re(H_s(f_c))<0 \end{cases} \quad (4.9)$$

where $H_s(f_c)$ is the component of the received signal path transfer function at the carrier frequency, f_c. Once this phase is found, the impulse response of the signal path must be transformed to reflect the passband demodulation. This transformation can be found as follows. We recall that the received passband signal r(t) can be expressed in terms of its equivalent baseband signal v(t) through the equation

$$r(t)=Re[v(t)e^{j2\pi f_c t}]$$
$$=v_r(t)\cos(2\pi f_c t)-v_i(t)\sin(2\pi f_c t) \quad (4.10)$$

where $v(t)=v_r(t)+jv_i(t)$. Now, to recover the information in this passband signal we must multiply it by the demodulating signal $e^{-j(2\pi f_c t+\theta_{del})}$ where θ_{del} is defined in Eqn.(4.9). By using the identities

$$\cos(u+v) = \cos u \cos v - \sin u \sin v$$
$$\sin(u+v) = \sin u \cos v + \cos u \sin v$$
$$\cos^2 u = 0.5\,(1 + \cos 2u)$$
$$\sin^2 u = 0.5\,(1 - \cos 2u)$$
$$\sin u \cos u = 0.5 \sin 2u$$

and temporarily letting $\chi=2\pi f_c t$, this demodulated signal $v_d(t)$ is given by

$$v_d(t) = [v_r(t)\cos(\chi)-v_i(t)\sin(\chi)]\,[\cos(\chi+\theta_{del})-j\sin(\chi+\theta_{del})]$$

$$= [v_r(t)\cos(\chi)-v_i(t)\sin(\chi)]\,[\cos(\chi)\cos(\theta_{del})-\sin(\chi)\sin(\theta_{del})]$$
$$-j[v_r(t)\cos(\chi)-v_i(t)\sin(\chi)]\,[\sin(\chi)\cos(\theta_{del})+\cos(\chi)\sin(\theta_{del})]$$

$$
\begin{aligned}
&- v_r(t)\,[\cos^2(\chi)\cos(\theta_{del}) - \cos(\chi)\sin(\chi)\sin(\theta_{del})] \\
&+ v_i(t)\,[-\cos(\chi)\sin(\chi)\cos(\theta_{del}) + \sin^2(\chi)\sin(\theta_{del})] \\
&- j\,(v_r(t)\,[\cos(\chi)\sin(\chi)\cos(\theta_{del}) + \cos^2(\chi)\sin(\theta_{del})] \\
&\quad - v_i(t)\,[\sin^2(\chi)\cos(\theta_{del}) + \cos(\chi)\sin(\chi)\sin(\theta_{del})])
\end{aligned}
$$

$$
\begin{aligned}
&- 0.5\, v_r(t)\,[(1+\cos(2\chi))\cos(\theta_{del}) - \sin(2\chi)\sin(\theta_{del})] \\
&+ 0.5\, v_i(t)\,[-\sin(2\chi)\cos(\theta_{del}) + (1-\cos(2\chi))\sin(\theta_{del})] \\
&- j\,0.5(v_r(t)\,[\sin(2\chi)\cos(\theta_{del}) + (1+\cos(2\chi))\sin(\theta_{del})] \\
&\quad - v_i(t)\,[(1-\cos(2\chi))\cos(\theta_{del}) + \sin(2\chi)\sin(\theta_{del})])
\end{aligned} \tag{4.11}
$$

Now, the square-root raised cosine filter in the receiver will eliminate all of the terms that have a $\cos(2\chi)$ or $\sin(2\chi)$. Hence, we may remove these terms in Eqn.(4.11) to produce the result

$$
\begin{aligned}
v_{df}(t) &= 0.5\,[v_r(t)\cos(\theta_{del}) + v_i(t)\sin(\theta_{del})] \\
&\quad - j\,0.5\,[v_r(t)\sin(\theta_{del}) - v_i(t)\cos(\theta_{del})]
\end{aligned} \tag{4.12}
$$

where the real and imaginary components of $v_{df}(t)$ are the real-valued baseband in-phase and quadrature components of the received signal after demodulation, respectively. From this equation, we see that the transformation which must be applied to the impulse response of the signal path is given by the expressions

$$
\begin{aligned}
h_{sin}(i) &= \mathrm{Re}(h_s(i))\,\frac{\cos(\theta_{del})}{2} + \mathrm{Im}(h_s(i))\,\frac{\sin(\theta_{del})}{2} \\
h_{squ}(i) &= \mathrm{Im}(h_s(i))\,\frac{\cos(\theta_{del})}{2} + \mathrm{Re}(h_s(i))\,\frac{\sin(\theta_{del})}{2}
\end{aligned} \tag{4.13}
$$

where i is an index of time, $h_s(i)$ is the complex-valued baseband signal path impulse response before demodulation and $h_{sin}(i)$ and $h_{squ}(i)$ are the real-valued baseband in-phase and quadrature components of the received signal impulse response after demodulation, respectively.

Hence, other than this transformation and the initial frequency translation of the channel's transfer function, the method of getting the signal path impulse responses for both the baseband and passband systems are the same.

We may now continue on and describe what to do with this signal path impulse response. We recall that this impulse response will have a resolution of 16 samples per symbol. This resolution results in a very nice looking continuous pulse when drawn on a computer. However, we are eventually going to perform a convolution with this impulse response and the larger the number of samples per symbol, the larger the computational time required. Hence, we would like to use as few samples per symbol as possible. Since this is an impulse response and we will be convolving it with a train of impulses representing a sequence of symbols, we could use just 1 sample per symbol. However, many of the equalizers in the transceivers use a forward T/2 fractionally spaced equalizer which requires 2 samples per symbol. As a result of this larger resolution being required in many of the simulations, 2 samples per symbol were used in all the simulations. It must be noted, however, that the train of impulses only has a resolution of 1 sample per impulse. In order to perform the convolution properly, we must insert zero samples between adjacent impulses in the train. This will effectively increase its resolution to 2 samples per symbol to match the resolution of the impulse response.

One last thing to consider when sampling the received signal path impulse response is the sampling phase. In the simulation, it was chosen in one of two ways. If the equalizer only had a forward filter with no feedback section then the cursor sample was selected as the peak

value of the impulse response. If, however, a feedback filter was used in the equalizer then the criterion used to select the cursor position was to choose the sampling phase which minimized the magnitude of the first (immediately preceding) precursor relative to the present cursor. This choice of sampling phase reduces the amount of precursor ISI that will be present in the received samples and will thus reduce the amount of noise enhancement caused by the forward equalizer. Should this choice of sampling phase result in an increase in the post-cursor ISI, the DFE will still eliminate this larger ISI with no noise enhancement. This latter choice of sampling phase did give an improvement in performance over simply choosing the peak value of the impulse response. Consequently, by choosing the appropriate sampling phase, by sampling the signal path impulse response at 2 samples per symbol and by truncating the response at the previously defined lengths, we can use this response as an FIR filter.

4.3.2 Echo Model

By recalling the generic transceiver structure of Figure 4.7, we see that the echo must pass through the square-root raised cosine transmit filter, the line driver, across the hybrid, through the square-root raised cosine receive filter and if it is a baseband system, it must also pass through the multiple response fixed pre-equalizer. After all of this , it is attenuated by the echo canceller before it enters the adaptive equalizer. All of these devices are again linear and can be lumped into one equivalent channel. This is also shown in Appendix F where the individual and the composite transfer functions of the echo paths (one for each side of the loop) are derived. From these composite

transfer functions, both the baseband and the passband impulse responses can be found in exactly the same way as was described for the received signal path impulse response. Each of the echo impulse responses will again initially have 16 samples per symbol but will be reduced to 2 samples per symbol as before. The timing phase of the echo impulse response, however, will always be chosen to sample at the peak value of the response. This will maximize the echo and will correspond to the worst case analysis. With this sampling rate and phase we can model this impulse response as another FIR filter.

4.3.3 Coloured Gaussian Noise Model

By again recalling the generic transceiver structure of Figure 4.7, we see that the white Gaussian noise must enter the transceiver through the hybrid and will then pass through the square-root raised cosine receive filter. If it is a baseband system, it will also pass through the multiple response fixed pre-equalizer. The combination of these devices produces a transfer function for the coloured Gaussian noise. This transfer function is derived in Appendix F. Both the baseband and the passband impulse responses for the coloured Gaussian noise can be found in exactly the same way as was described for the received signal's impulse response. The coloured Gaussian noise impulse response will again have 16 samples per symbol. However, this impulse response will be convolved with samples of a white Gaussian noise waveform. Consequently, the minimum number of 1 sample per symbol may no longer be sufficient. The number of samples per symbol is now dictated by the sampling theorem which tells us that we must sample a signal at at least twice its maximum frequency for no aliasing to occur.

We know our coloured Gaussian noise is going to have its maximum frequency limited to the bandwidth of the square-root raised cosine filter. If 100% excess bandwidth pulses are used, then the bandwidth of the filter will be 1/T Hz. As a result, this will be the maximum frequency content of the coloured noise. The sampling theorem again tells us that we must sample the noise at at least twice this maximum frequency or a rate of 2/T samples per second. Since there are 1/T symbols per second, this implies that we must sample the coloured Gaussian noise at at least $(2/T)\cdot(T/1) = 2$ samples per symbol. Behold, we were already using this number of samples per symbol because of the equalizers and thus, this added constraint does not cost us any increased complexity. Consequently, we may afford to sample the impulse response only 2 times per symbol without having any aliasing problems. The sampling phase is again chosen so that the cursor sample is taken at the maximum value of the impulse response. This will give us our third FIR filter.

4.3.4 Near-end Crosstalk Model

By once again recalling the generic transceiver structure of Figure 4.7 and viewing the simulated signal processing structure of Figure 4.20, we see that the NEXT must pass through the square-root raised cosine transmit filter, the line driver, a hybrid, the NEXT coupling transfer function, a second hybrid, a square-root raised cosine receive filter and if it is a baseband system, it must also pass through the multiple response fixed pre-equalizer. These individual transfer functions along with the composite transfer functions of the NEXT paths (one for each side of the loop) are shown in Appendix F. Once again,

both the baseband and passband impulse responses can be found by following the exact steps described for the received signal impulse responses. This will produce a NEXT impulse response with 16 samples per symbol. As was mentioned in section 2.2.4, the NEXT will be modelled as having a Gaussian distribution. Consequently, this impulse response will also be convolved with samples of a white Gaussian noise waveform. From the arguments given for the coloured Gaussian noise impulse response, we may sample the NEXT impulse response at 2 samples per symbol. The sampling phase was again chosen to have the cursor sample correspond to the maximum value of the impulse response. This will give us our fourth and final FIR filter.

4.3.5 Impulse Noise

The one noise source which was considered important in Chapter 2, and is not modelled in the simulation, is impulse noise. Impulse noise can have a significant impact on the performance of a transceiver operating on the central office side of the loop. Unfortunately, the only available model for impulse noise, given in [22], is difficult to apply in this type of simulation. The reason for this is that impulse noise, by its very definition, consists of infrequent noise events. In the model, we might have an impulse noise event occur every 5 to 10 minutes in real-time. However, in each run of our simulation, we transmit a maximum of 10^6 bits which at 800 kb/s only represents 1.25 seconds of real-time. Consequently, if an impulse noise event is included in these 1.25 seconds, the simulation will produce a badly skewed bit-error-rate (BER). One possible solution to this problem is to run two different simulations. One simulation would be with an

impulse noise event and one simulation would be without. We could then weigh the two resulting BERs, in some manner, to get an estimate of the system's true BER when impulse noise is a disturbance. However, it could then be argued that a number of simulations, using different amplitude impulse noise events, should be run and have all of their BERs included in the weighting. This does not seem to be a very practical solution. As a result, impulse noise was not modelled at all in this simulation. One consolation is, however, that it is NEXT which is considered the dominant source of noise in a digital subscriber loop system.

4.3.6 Generation and Scaling of the Signal and Echo Impulses

Now that we have the received signal and echo path impulse responses, we can examine how to generate a train of impulses which will represent a sequence of symbols. The first step is to use a random bit generator to produce a series of bits. If no trellis encoding is used then these bits are mapped directly into symbols using one of the mappings shown in Figure 4.8, 4.12 or 4.16 depending on the system being simulated. When trellis encoding is used the generated bits are first convolutionally encoded before they are mapped. This is done by implementing the convolutional encoders exactly as they are shown in the previous figures through the use of modulo-2 adders and symbol spaced delays. Once the bits are encoded they are mapped to symbols using one of the mappings shown in Figures 4.9, 4.13 or 4.17.

At this point, it should be noted that the symbol values are not the absolute amplitudes which are transmitted. These symbols must be scaled to ensure that the transmitted signal's power, at some reference

point, meets a specific standard. As was previously mentioned, the basic access interface standard specifies that the average power at the line and network interfaces should be at 13.5 dBm over a 135 ohms test resistance. Since we wanted to compare the performance of the basic access transceiver to other more complicated transceivers, we used this transmitted power level and reference point in all the baseband and passband transceivers.

To obtain an average power of 13.5 dBm at the line and network interfaces, two scaling factors must be calculated. The first factor scales the symbol values so that they themselves would have the correct amplitudes to produce a power of 13.5 dBm over a 135 ohms resistance. The second factor compensates for the voltage division which occurs within the hybrid circuit as shown in Appendix F. By using both of these factors together a signal power of 13.5 dBm at the line and network interfaces is assured.

The first factor can be calculated as follows. If the symbols of a baseband system are equiprobable, then the variance or power associated with a scaled sequence of these symbols is given by

$$\sigma^2 = \frac{1}{M_b} \sum_{m=1}^{M_b} (a_m \cdot A)^2 = \frac{A^2}{M_b} \sum_{m=1}^{M_b} (a_m)^2 = \frac{(M_b^2-1)A^2}{3} \qquad (4.14)$$

where M_b is the number of symbols in the baseband alphabet, A is the scaling factor and the values of a_m are given in Figures 4.8 and 4.9. For the 4 and 8 PAM signals, their variances can be shown to equal $\sigma^2 = 5A_{4p}^2$ and $21A_{8p}^2$, respectively. The power of a passband signal which is represented by an equivalent complex-valued baseband signal is given by

$$\sigma^2 = \frac{1}{2M_Q} \sum_{m=1}^{M_Q} |a_m \cdot A|^2 = \frac{A^2}{2M_Q} \sum_{m=1}^{M_Q} |a_m|^2 \qquad (4.15)$$

where M_Q is the number of symbols in the complex-valued baseband alphabet and the values of a_m are given in Figures 4.12, 4.13, 4.16 and 4.17. For the 8, 16 and 32 QAM signals, their variances can be shown to equal $\sigma^2 = 5A_{8Q}^2$, $5A_{16Q}^2$ and $10A_{32Q}^2$, respectively. These powers over a 135 ohms test resistance, can be expressed in dBm through the relationship.

$$P_{av}(\text{dBm}) = 10 \cdot \log_{10}(\sigma^2 \cdot 1000/135) \qquad (4.16)$$

Since P_{av} has been specified to equal 13.5 dBm, the variance of a sequence of scaled symbols should equal $\sigma^2 = 3.022$. This value for σ^2 results in the scaling factors $A_{4P} = .777$, $A_{8P} = .379$, $A_{8Q} = .777$, $A_{16Q} = .777$ and $A_{32Q} = .550$.

By applying these scaling factors, all of the symbol sequences generated will have a power of 13.5 dBm. However, when these signals are passed through the square-root raised cosine transmit filter and the hybrid to get to either the line or network interface, their powers will be reduced because of the filtering and the voltage division which occurs. The second scaling factor is used to compensate for this loss of power. Since the voltage division at the hybrid will be frequency dependent, the best way of finding this second scaling factor is to send a test pulse through the filter and hybrid and find how its power is affected. This was done in the simulation. The composite transfer function of the filter and hybrid is shown in Appendix F. Two opposite polarity impulses were sent through this transfer function and the input and output signal power were calculated. This ratio of input to output power will tell us exactly how much of the signal power is missing and

will thus give us our second scaling factor. The second scaling factor will simply be

$$B=(P_{in}/P_{out})^{0.5}, \tag{4.17}$$

where the square-root is necessary because we are scaling the signal voltage.

When we combine the two scaling factors the result will equal the gain of the line driver and is given by

$$G=A(P_{in}/P_{out})^{0.5} \tag{4.18}$$

The scaling factor A can take on any of its previously defined values depending on the system being simulated. The values of P_{in} and P_{out} will depend on the particular subscriber loop and the transmission rate used. It should be noted, that unless the subscriber loop is symmetric, the value of G on the two sides of the loop will be different. This is because the impedance of the loop as seen from the hybrid will be different on the loop's two sides and will thus cause a different voltage division to occur.

This scaling ensures that both the signal and echo are at their correct power levels. The way the coloured Gaussian noise and NEXT samples are generated and scaled are discussed in the following sections.

4.3.7 Generation and Scaling of Coloured Gaussian Noise

The coloured Gaussian noise at the receiver is obtained by convolving samples of a white Gaussian noise waveform with the coloured Gaussian noise impulse response. In the baseband systems, the white Gaussian noise samples are generated by a zero-mean unit variance Gaussian number generator. In the passband systems, these samples are

again generated by a zero-mean Gaussian number generator, except they are complex-valued and both the real and imaginary parts have a unit variance. Once the samples are generated they must be scaled to their proper power levels before they can be used. The scaling factor for the coloured Gaussian noise can be found by calculating the total power across the 135 ohms load resistance in the hybrid. The expression for the total normalized white Gaussian noise power (over a 1 ohm load resistance) is given by

$$P_{pu} = \int_0^{BW} N_0 \, df \tag{4.19}$$

where N_0 is the single sided (positive frequencies) flat power spectral density and BW is the bandwidth of the noise. In this case, the bandwidth of the white noise is constrained to the maximum frequency which can be represented by the FFT. This is given, with the use of Eqn.(4.8), by

$$f_{max} = \frac{\Delta f \cdot NN}{2} = \frac{1}{NS \cdot T} \frac{NS \cdot SS}{2} = \frac{SS}{2T} \tag{4.20}$$

where SS=16 is the number of samples per symbol in the 2048 point data array. Using $BW=f_{max}$, the total (denormalized) white Gaussian noise power over a resistor R_L is given by

$$P_{WGN} = N_0 f_{max} R_L = 8 N_0 R_L / T \tag{4.21}$$

where, in these simulations, $N_0 = 5*10^{-14}$ W/Hz and R_L=135 ohms. Since the white Gaussian noise samples only have a power of 1, the voltage scaling factor for a baseband system is the square-root of the expression given by Eqn.(4.21) which is

$$CGN_{BB} \text{ scaling factor} = (8 N_0 R_L / T)^{0.5} \tag{4.22}$$

The coloured Gaussian noise scaling factor used in the passband systems is very similar to Eqn.(4.22). However, since the spectra of the passband signals are shifted down in frequency to be centred at d.c., the complex-valued white Gaussian noise has a power spectral density of N_0 over both positive and negative frequencies. Consequently, there is an additional factor of two inside the square-root of Eqn.(4.22). This results in the coloured Gaussian noise voltage scaling factor for a passband system to be given by

$$CGN_{PB} \text{ scaling factor} = (16N_0R_L/T)^{0.5} \qquad (4.23)$$

4.3.8 Generation and Scaling of NEXT

The NEXT is also generated by convolving samples of a white Gaussian noise waveform with the impulse response of the NEXT path. Consequently, the same Gaussian number generators which were described previously are used here. The scaling of these white Gaussian noise samples to produce the proper NEXT power levels is, however, quite different from the scaling performed for the coloured Gaussian noise.

We recall that the NEXT is caused by disturbing near-end signals transmitted over adjacent loops, and capacitively coupled to the disturbed subscriber loop. Since the power coupling transfer function, $K_{NEXT}f^{1.5}$, takes into account the coupling, the sources of the NEXT are simply the other transmitted signals which all have a power of 13.5 dBm at the line interface. As a result, the voltage scaling of the NEXT should be done in a manner which will produce this same power of 13.5 dBm at the line interface before the multiplication by the coupling transfer function. The composite transfer functions of the NEXT paths (Eqns.(F.14) and (F.15)) already contain the gains of the line drivers

defined in Eqn.(4.18). As we recall, these gains were chosen to ensure that a sequence of unscaled symbols would produce a power of 13.5 dBm at the line interface. Hence, all we have to do is to scale the white Gaussian noise samples so that their variance is equal to the variance of the particular unscaled symbols which are going to be transmitted. Therefore, the NEXT voltage scaling factor for both the baseband and passband systems is given by σ, where σ^2 can be obtained from Eqns.(4.14) and (4.15) by setting A=1.

4.4 Adaptive Equalizers and Viterbi Decoder

Once all of the symbol and noise sequences have been convolved with their appropriate impulse responses, there will be a sequence of noisy samples at the receiver which will have to be equalized. All of the adaptive equalizers simulated use the least mean squares (LMS) adaptation algorithm to update their taps. However, as was discussed in section 2.1.3.3.2, the equalizers must be trained before they can effectively reduce the ISI caused by the channel.

The length of the training period is a very important consideration in the simulation. If it is chosen to be too short, the equalizer's coefficients will be far from their optimum values when it begins operation in the decision-directed mode. This may cause a substantial number of bit errors at the beginning of the transmission, which will make the system's performance seem poorer than it really is.

The first method used in the simulations to determine if the taps had converged was based on measuring the amount of change in the value of the forward equalizer's cursor tap. An average value for this tap was calculated every 1024 symbols. If the present average and the last

average were different by less than 1%, then the equalizer's taps were considered to have converged. This method was moderately successful.

The second method, which was eventually used, was to simply get a feel for the length of time it took an equalizer to converge and "manually" set the training period at this length. This was usually done by examining the averages of the first method and perhaps running a short test run. A typical value for the training length which assured a good equalizer convergence was 20,480 symbols.

The equalizers and the Viterbi decoder (or the parallel decision feedback decoder) were simulated exactly as they were described in the previous chapters. For each of the adaptive equalizers, an attempt was made to optimize its step size μ. As a result, the step sizes used in the simulations ranged from 0.0001 to 0.0008 depending on the specific equalizer and the type of coding and modulation used in the transceiver.

The truncation depth of the Viterbi decoder was usually chosen to be 30 symbols. This depth would be slightly long for most loops, however, for worst case loops with strong noise and residual ISI after equalization this depth was needed to ensure the surviving paths merged into one path. If a Viterbi decoder was not used (because there was no trellis encoding), then the symbol decisions were made directly on the equalizer's output. The criterion used to make these decisions was to simply choose the symbol which was closest in Euclidean distance to the equalized sample. After the symbol decisions were made, the symbols were decoded to produce blocks of bits. These received bits were then compared to the far-end transmitted bits to determine if any decision errors had been made. By counting the total number of errors made in

both directions of transmission, we could then determine the system's performance by calculating its bit error rate on both sides of the loop.

4.5 Generation of System Performance Measurements

4.5.1 Quantizer Signal to Noise Ratio

In this simulation, the first system performance estimates made by the program were calculations of the quantizer signal to noise ratio (QSNR) at both transceivers. The QSNR is calculated using the received samples that are just about to enter the decision making device. These samples have already passed through all of the receiver's filters including any fixed or adaptive equalizers. As a result, the QSNR will reflect any noise enhancement caused by these equalizers trying to eliminate or reduce the ISI on the signal. This makes the QSNR a valuable tool in estimating the performance of different equalizer structures. The QSNR can be calculated using the expression

$$QSNR = 10 \cdot \log_{10} \left[\frac{1}{N} \sum_{n=1}^{N} |a_n|^2 / |a_n - y_n|^2 \right] \qquad (4.24)$$

where a_n are the desired symbols and y_n are the equalized samples just before the decision making device. In these simulations, the QSNR was calculated by averaging the signal to noise power over the last 5120 symbols of the training period. This is another reason why the length of the training sequence must be adequately long.

4.5 2 Bit Error Rate Estimation

The second and most important measure of a system's performance is its bit-error-rate (BER). The Monte Carlo method of estimating BER was

used in these simulations. This is a brute force technique which makes no a priori assumptions about a system. We simply transmit bits or symbols and count how many errors are made at the receiver. For instance, if N bits are transmitted and n errors are detected, then the estimate of the BER is given by

$$\bar{p}=n/N \tag{4.25}$$

As the number of transmitted bits $N\to\infty$, the estimate of the BER, $\bar{p}$, will converge to its true value. However, for a finite number of transmitted bits, we will want to know how good this BER estimate is. We can answer this question by defining a confidence interval h_1-h_2 where h_1 and h_2 are BERs which lie on either side of $\bar{p}$. That is, $h_2\leq\bar{p}\leq h_1$. We can then specify a certain confidence level ζ that $\bar{p}$ is between h_1 and h_2. This gives the expression

$$\text{Prob}(h_2\leq\bar{p}\leq h_1)=\zeta \tag{4.26}$$

Figure 4.21 [36] shows confidence intervals, h_1-h_2, for the three confidence levels of 90, 95 and 99%. From these curves, a rule of thumb is obtained which states that the number of transmitted bits N should be on the order of $10/\bar{p}$. This will give a confidence interval of about $(h_1=2\bar{p},h_2=0.5\bar{p})$ with a 97% confidence level. In our simulations, we wanted BERs as low as 10^{-5}. Consequently, for these runs, we should transmit at least 10^6 bits so that our estimate of the BER has at least this reliability.

As a summary of this chapter, it is worthwhile examining the general flowchart of Figure 4.22. This flowchart illustrates the sequence of steps that were taken in each of the programs which simulated a different transceiver structure. The programs began by reading in and initializing various constants. These constants included

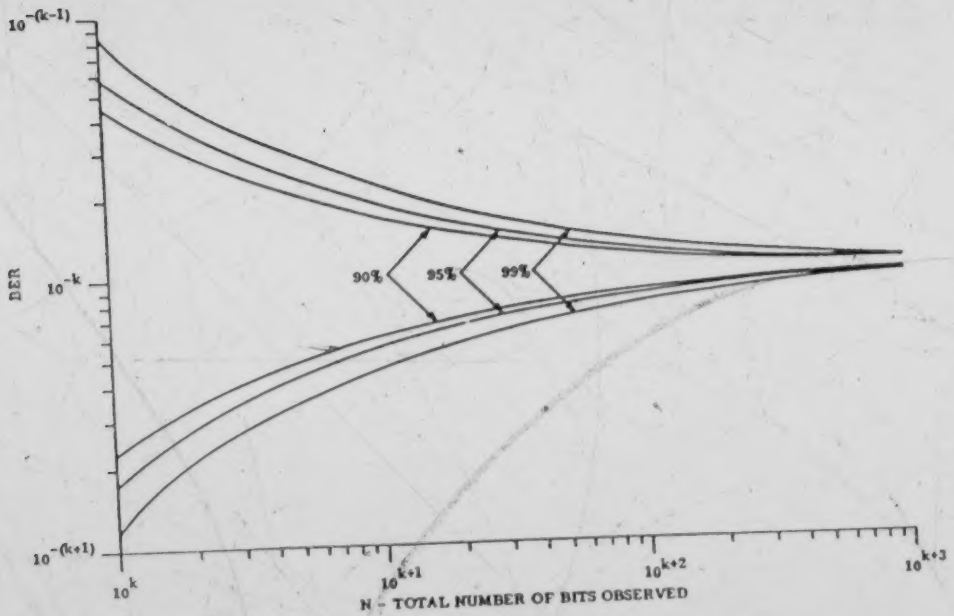


Figure 4.21 Confidence bands on BER when observed value is 10^{-k}

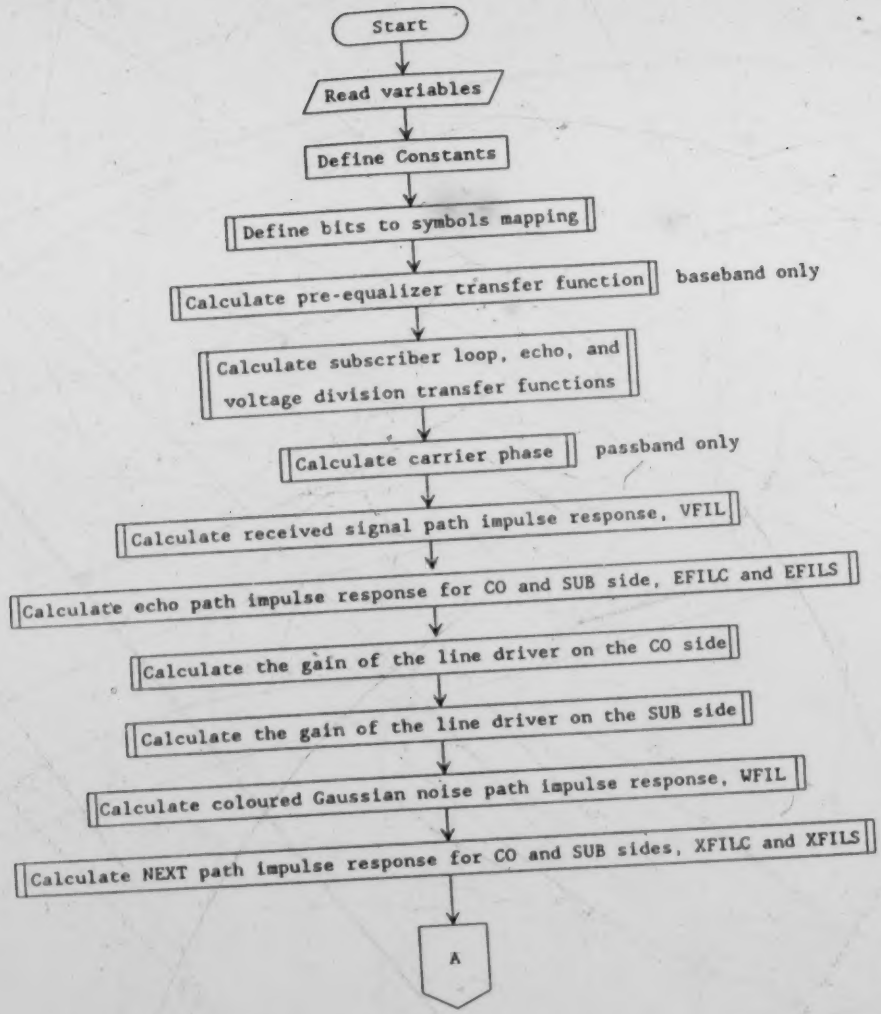

Start
Read variables
Define Constants
Define bits to symbols mapping
Calculate pre-equalizer transfer function
baseband only
Calculate subscriber loop, echo, and voltage division transfer functions
Calculate carrier phase
passband only
Calculate received signal path impulse response, VFIL
Calculate echo path impulse response for CO and SUB side, EFILC and EFILS
Calculate the gain of the line driver on the CO side
Calculate the gain of the line driver on the SUB side
Calculate coloured Gaussian noise path impulse response, WFIL
Calculate NEXT path impulse response for CO and SUB sides, XFILC and XFILS
A

Figure 4.22 General flowchart of programs simulating digital subscriber loop system

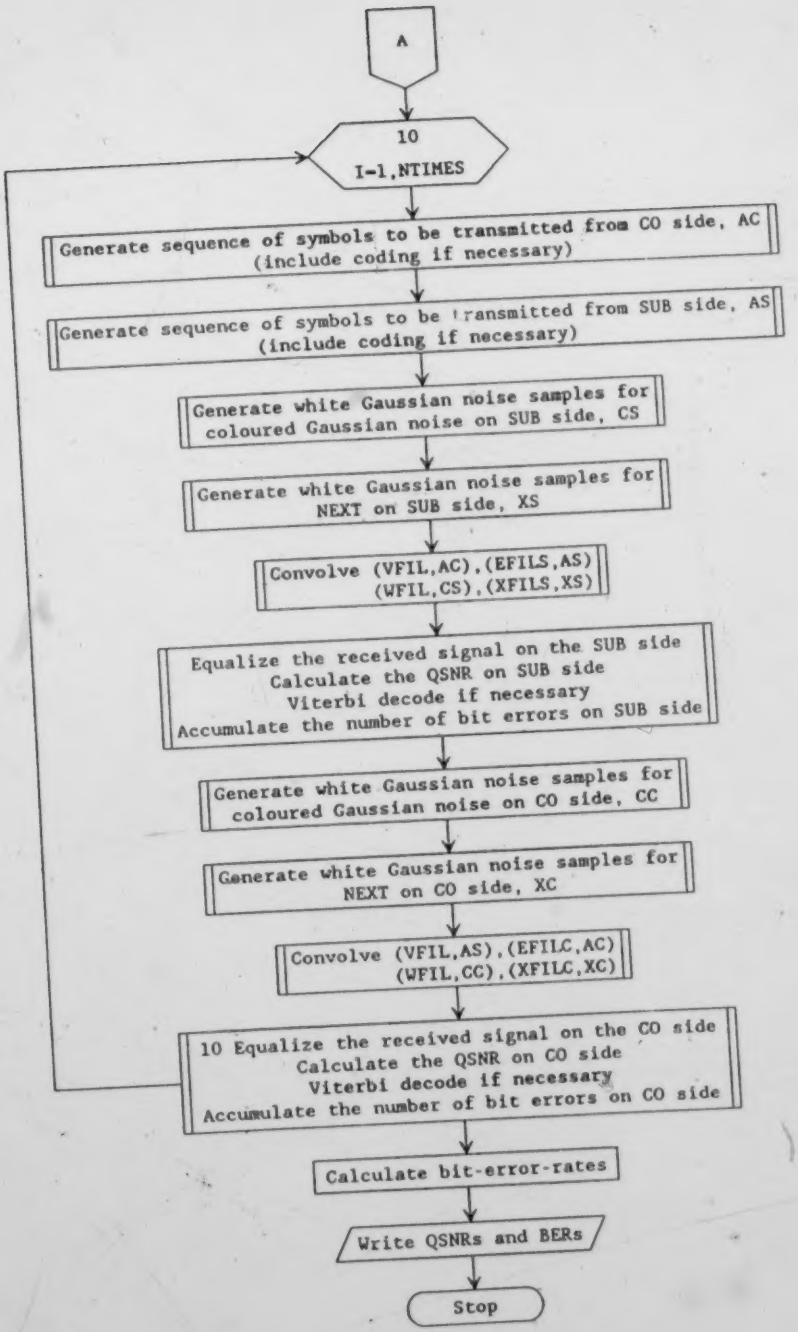

A
10
I-1,NTIMES
Generate sequence of symbols to be transmitted from CO side, AC
(include coding if necessary)
Generate sequence of symbols to be transmitted from SUB side, AS
(include coding if necessary)
Generate white Gaussian noise samples for
coloured Gaussian noise on SUB side, CS
Generate white Gaussian noise samples for
NEXT on SUB side, XS
Convolve (VFIL,AC),(EFILS,AS)
(WFIL,CS),(XFILS,XS)
Equalize the received signal on the SUB side
Calculate the QSNR on SUB side
Viterbi decode if necessary
Accumulate the number of bit errors on SUB side
Generate white Gaussian noise samples for
coloured Gaussian noise on CO side, CC
Generate white Gaussian noise samples for
NEXT on CO side, XC
Convolve (VFIL,AS),(EFILC,AC)
(WFIL,CC),(XFILC,XC)
10 Equalize the received signal on the CO side
Calculate the QSNR on CO side
Viterbi decode if necessary
Accumulate the number of bit errors on CO side
Calculate bit-error-rates
Write QSNRs and BERs
Stop

Figure 4.22 General flowchart of programs simulating digital subscriber loop system (Con't)

the roll-off factor, the baud rate, the carrier frequency if applicable, the echo suppression level, the number of taps in the equalizer, the step size for the equalizer's adaptation algorithm, the length of the training sequence and the truncation depth of the Viterbi decoder, if one was necessary. Following this, the specific bit to symbol mapping that was to be used for a particular transceiver was defined. The program would then jump into calculating all of the different transfer functions needed to describe the system. These are the transfer functions which were just discussed in sections 4.3.1 to 4.3.4 . Once these transfer functions were known, the program could then calculate their associated impulse responses. Included within this segment of the program were the calculations of the voltage scaling factors required in the simulations. Up to this point, the program is essentially in its initialization stage.

Next, the program enters into its transmission stage. Here, sequences of symbols and samples of noises are generated and then convolved with their appropriate impulse responses. This is done for one direction of transmission at a time and in blocks which will produce 1024 received samples. Once a block of received samples is generated, these samples will be equalized and then used to calculate the QSNR on that particular side of the subscriber loop. These samples will then enter the decision making device or the Viterbi decoder if trellis coding is used. Once a symbol decision is made, the symbol is reverse mapped into received bits. These bits will be compared to the known transmitted bits to determine if any decision errors were made. When the program leaves its transmission stage, it will know the QSNR on both sides of the loop, the number of bits sent from each transmitter and the

number of errors made at each receiver. The program then enters its output stage when it simply calculates the BER for both directions of transmission, and writes the QSNRs and the BERs for both sides of the digital subscriber loop to a file.

Chapter 5

Computer Simulation Results

In the first two chapters of this thesis, an introduction to both subscriber loops and to the main impairments of digital transmission over these loops was given. While keeping the transmission impairments in mind, a discussion of a number of modulation and coding techniques which seemed well suited to this noisy and signal corrupting environment followed. This led to the various transceiver structures which were modelled as described in Chapter 4 and eventually simulated by writing several different FORTRAN programs. Finally, in this chapter, the simulation results which were obtained from running these programs will be presented and discussed.

Due to the long run times that were required to produce BERs on the order of 10^{-5}, a great deal of effort was placed in finding a suitably fast computer environment that could satisfy our demands. Initially, a BEST 386 computer with an 80387 math co-processor and an Eighteen Eight Laboratories PL1250 Floating Point Array Processor was used to run the programs. This array processor is based on an AT&T 32-bit digital signal processor which has a processing rate of 12.5 million floating point operations per second. It operates in parallel with the host PC and performs computations on arrays and matrices. This was a perfect solution to the four convolutions which were necessary for each direction of transmission. For this section of the program, the array processor increased the computers speed by a factor of 100 which made the time spent on these operations negligible. However, the adaptive equalization (and Viterbi decoding) portion of the program had to operate on a symbol by symbol basis. This eliminated the effectiveness

of the array processor. In total, the array processor reduced some run times by as much as a factor of 8 and some run times by as little as a few percent. Where a particular simulation would fit into this broad range was simply a function of how much work could be done by the array processor. Unfortunately, it was the longest runs which were reduced by the few percent. Consequently, with run times as long as 72 hours or more per simulation (that is one point on a graph) we began to run the programs in parallel using up to 13 or 14 IBM-AT compatibles and SUN 3-50 and 3-60 work stations. Although this involved a great number of file movements, this is where most of the simulation results were generated. Towards the end of the project, simulations were run on two newly acquired DECstation 3100 work stations. With computational speeds 7 times greater than the BEST 386 without the array processor, and no file transfers, this is the computer environment of choice.

Simulations were performed over two different subscriber loops, shown in Figure 5.1. These loops were chosen with the purpose of investigating which transceiver structures, if any, could provide the proposed high-bit-rate (800 kb/s) digital subscriber loop access. The first loop in the figure represents the farthest reaching loop in the Carrier Serving Area (CSA). The second loop was designed to be the worst case loop within the same CSA. It has the maximum length possible while having two equal length bridged taps near the subscriber side. The simulation results performed on these two loops will now be presented.

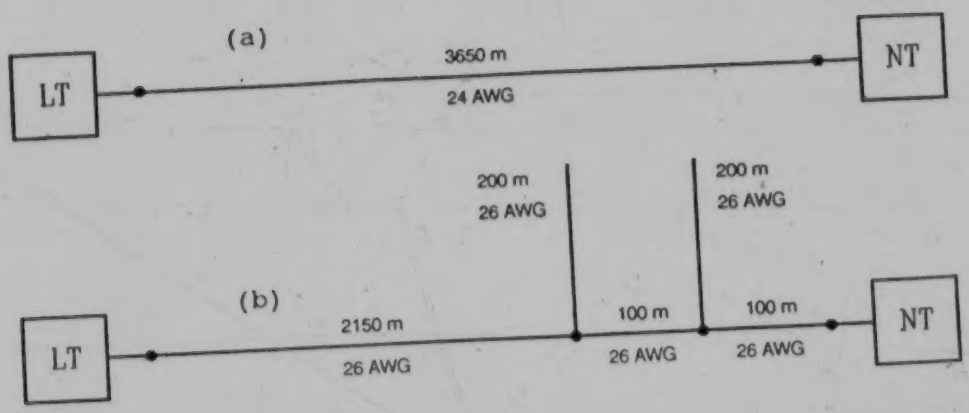


Figure 5.1 (a) Configuration of farthest reaching loop
(b) Configuration of worst case loop

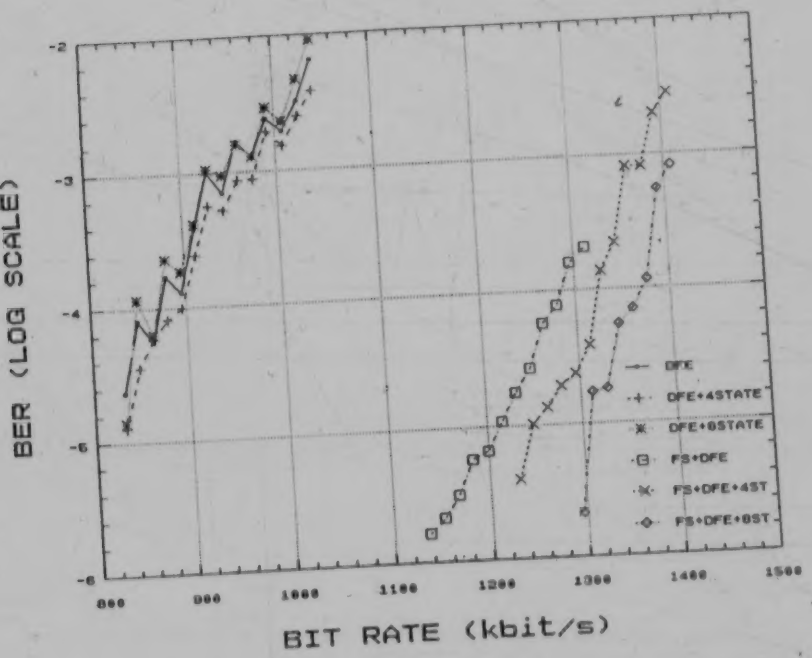


Figure 5.2 BER versus bit rate curves for the baseband transceivers transmitting on the loop in Figure 5.1a

5.1 Results on the Farthest Reaching Loop

5.1.1 Baseband Transceivers

We will begin the discussion of the results by examining the performance of the baseband transceivers when transmitting on the 3650m, uniform 24 AWG cable, shown in Figure 5.1a. A set of BER versus bit rate curves for this loop is presented in Figure 5.2. Since the loop is homogeneous BERs in both directions of transmission are the same and thus only one curve per transceiver is shown in the Figure. An echo cancellation level of 40 dB was chosen in addition to the isolation provided by the hybrid. This level is much smaller than the 65 dB reduction which can be provided by a state-of-the-art echo canceller, but the echo power for this loop, was already the smallest noise source by an order of magnitude.

The first observation that can be made is that there are two widely separated groups of curves. The group on the left shows the performance of the transceivers using only the fixed pre-equalizer in front of the 16 tap DFE. The group on the right illustrates the increased performance when the adaptive FSLE was placed in series with the pre-equalizer and the DFE. From this Figure, we see that through improved equalization alone, a gain of 400 kb/s at a BER of 10^{-5} is possible. This is evidence of the significance of having a T/2 fractionally spaced forward filter to reduce precursor ISI and act as a matched filter.

A second observation is that when a simple DFE structure is used, trellis coding gives no significant gain in performance over an uncoded system. However, when the more sophisticated equalizer was implemented, an improvement was found. The transceiver using the 4-state trellis

code gave a gain of approximately 40 kb/s, while the transceiver using the 8-state trellis code exhibited a gain of a 100 kb/s, over the uncoded system at a BER of 10^{-5}. The relatively poor performances of the coded systems, using the simple DFE structure, can be explained by recalling that these codes were designed for ISI free channels with additive white Gaussian noise. If the simple DFE structure is leaving a signficant amount of ISI on the signal (as seems to be the case) then the reduction of the free Euclidean distance which came with the expansion of the signal's constellation is not being more than compensated for by the true gain of the code in the presence of ISI. In addition to this, the essentially Gaussian noise samples at the input to the Viterbi decoder are coloured and are not white. This implies the noise samples will be correlated which could cause an increase in the probability of having several relatively large noise samples occur consecutively. This, in turn, could cause the Viterbi decoder to choose an incorrect path in the trellis more frequently. The performance gains obtained from trellis encoding while using an adaptive FSLE with a DFE again shows that this equalizer is doing a much better job of reducing ISI. It is also noted that the slopes of the BER curves for the trellis coded systems are slightly steeper than the curve for the uncoded scheme. This would result in even greater improvements in performance, then those stated above, when operating at lower BERs. As a final note to Figure 5.2, transmission at 800 kb/s and a BER of 10^{-7} would be possible with any of the transceivers using both the T/2 FSLE and the DFE. However, this transmission rate would not be possible with the simpler stand alone DFE structure.

5.1.2 Passband Transceivers

The next results we will discuss, illustrate the performance of the passband transceivers on the 24 AWG cable. Figure 5.3 shows a set of BER versus bit rate curves for the uncoded 8 AMPM and coded 16 QAM transceivers. Below it, Figure 5.4 shows a set of corresponding curves for the uncoded 16 QAM and coded 32 QAM transceivers. We first observe that in both figures there are again two widely separated groups of curves. The groups on the left, in both figures, illustrate the performance of the transceivers using only T and T/2 spaced linear equalization. The groups on the right show the increased performance when both a T/2 FSLE and a DFE are used together. We see that the improved equalization increases the performance of the uncoded systems in both figures by approximately 240 kb/s at a BER of 10^{-5}. This gain in performance can be attributed to the fact that the DFE with the T/2 FSLE will cause significantly less noise enhancement than just the linear equalizers.

A second interesting observation is the relative performance of the 21 tap symbol spaced linear equaliser and the 21 tap T/2 fractionally spaced linear equalizer. As stated in section 2.1.3.1.2, a T/2-spaced equalizer with the same number of taps as a T-spaced equalizer will span half the time interval. Nevertheless, our results confirm Qureshi's findings [15] that when both types of equalizers have the same number of taps, the FSLE will perform at least as well as the T-spaced equalizer on relatively simple channels such as this particular subscriber loop.

Further observations can be made by examining the gains and losses that come from trellis encoding. When just a linear equalizer is used,

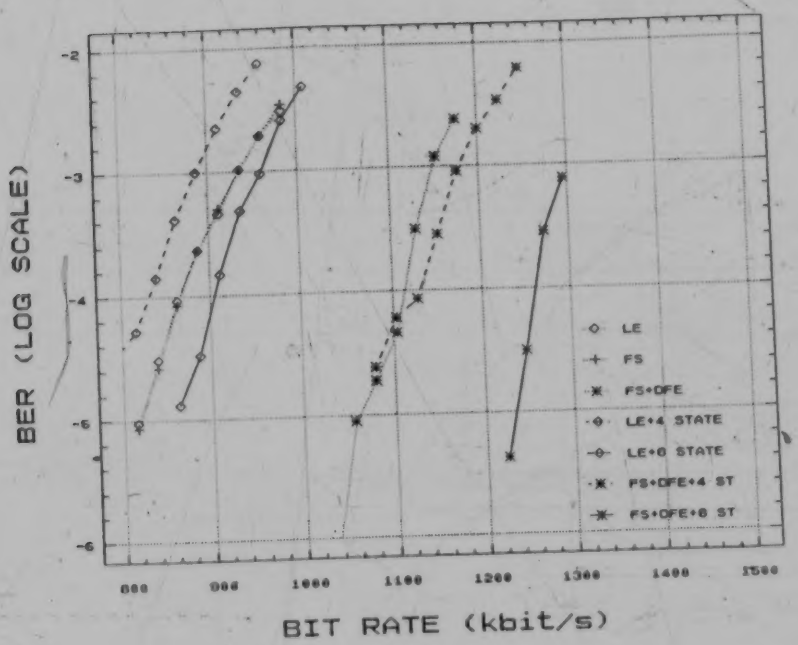


Figure 5.3 BER versus bit rate curves for the uncoded 8 AMPM and coded 16 QAM passband transceivers transmitting on the loop in Figure 5.1a

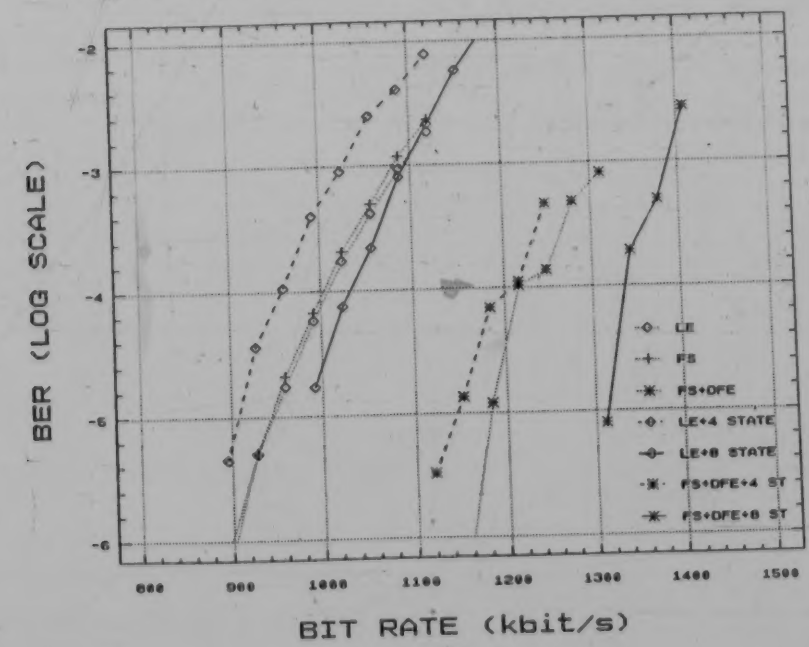


Figure 5.4 BER versus bit rate curves for the uncoded 16 QAM and coded 32 QAM passband transceivers transmitting on the loop in Figure 5.1a

the 8-state codes give small gains in performance while the 4-state codes display performance which is quite a bit poorer than that of the corresponding uncoded systems. When the better T/2 FSLE with a DFE section is used, we find that the 4-state codes again give no improvement in performance. However, the 8-state 16 QAM and 32 QAM codes both provide a significant increase in transmission rate of approximately 180 kb/s and 140 kb/s over their respective uncoded systems. The cases where trellis coding caused a reduction in performance instead of an increase can be explained by again considering the detrimental affects residual ISI and non-white noise can have on the true gains of the codes.

Perhaps the most important result found here, in terms of deciding which transceivers are best suited for the digital subscriber loop environment, can be seen by comparing the relative performances of the passband transceivers in the two figures. The trellis coded 32 QAM and uncoded 16 QAM transceivers outperformed the trellis coded 16 QAM and uncoded 8 AMPM transceivers, respectively. For each combination of equalizer and signal mapping the former transceivers consistently provided a transmission rate which was approximately 100 kb/s greater than the rate achievable on the latter transceivers.

When we now compare the performance of the best baseband transceiver to the best passband transceiver, we find their achievable bit rates are almost identical at 1300 kb/s at a BER of 10^{-5}. However, the passband transceivers are considerably more complicated. Thus, we can begin to lean towards the baseband solution. BER versus bit rate for another loop may help us decide the fate of the passband transceivers.

5.2 Results on the Worst Case Loop

Figure 5.5 illustrates the performance of the baseband transceivers on the loop shown in Figure 5.1b. Since this loop is not symmetric, the channel capacity in the two directions of transmission will be different. The curves labelled "CO" give the BERs on the central office's side, while those labelled "SUB" give the BERs on the subscriber's side. For the simulations on this loop, an echo cancellation of 60 dB was chosen. This larger level of cancellation can be provided by a state-of-the-art echo canceller and will be needed as the echo power is signicantly larger on this loop than on the former.

As shown in Figure 5.5, the performance of the simple DFE structure is still quite poor compared to the transceivers using both a forward FSLE and a DFE. One significant change, when comparing Figure 5.5 to Figure 5.2, is that the transceiver using the 4-state trellis code no longer experiences any gain in performance. This is due to the fact that the ISI will be worst on this loop because of the bridged taps. Hence, the residual ISI left on the signal after equalization is now large enough to deteriorate the code's gain so that there is actually a net loss in performance with coding. However, the transceiver using the 8-state trellis code has still maintained a significant gain of approximately 100 kb/s over the uncoded system.

The performances of the passband transceivers using the uncoded 16 QAM and the trellis coded 32 QAM signals on this second loop are shown in Figure 5.6. Once again we find there is a significant difference (180 kb/s) between the performances of the transceivers using different equalizers. We also see that the 8-state trellis code provides a gain of about 80 kb/s over the uncoded transceiver using the same equalizer.

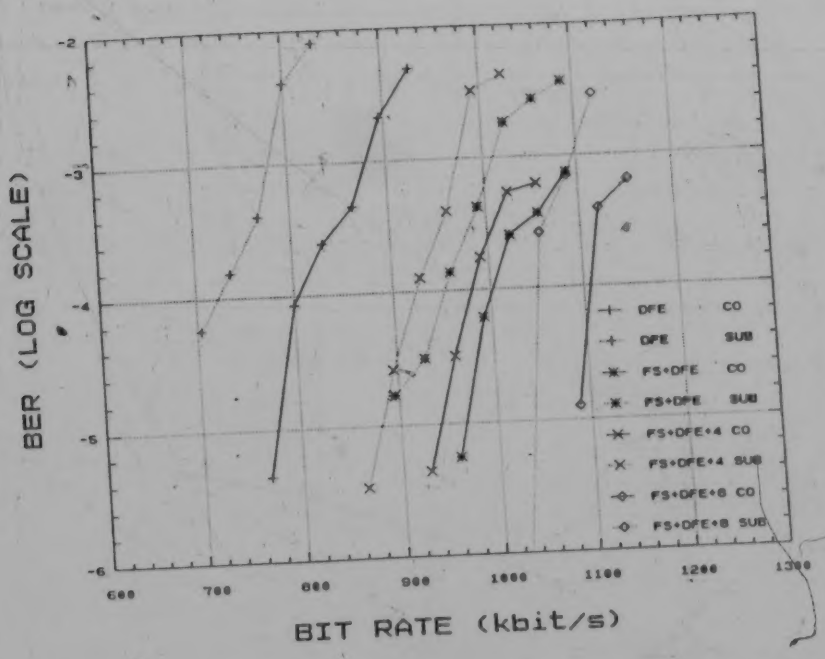


Figure 5.5 BER versus bit rate curves for the baseband transceivers transmitting on the loop in Figure 5.2a

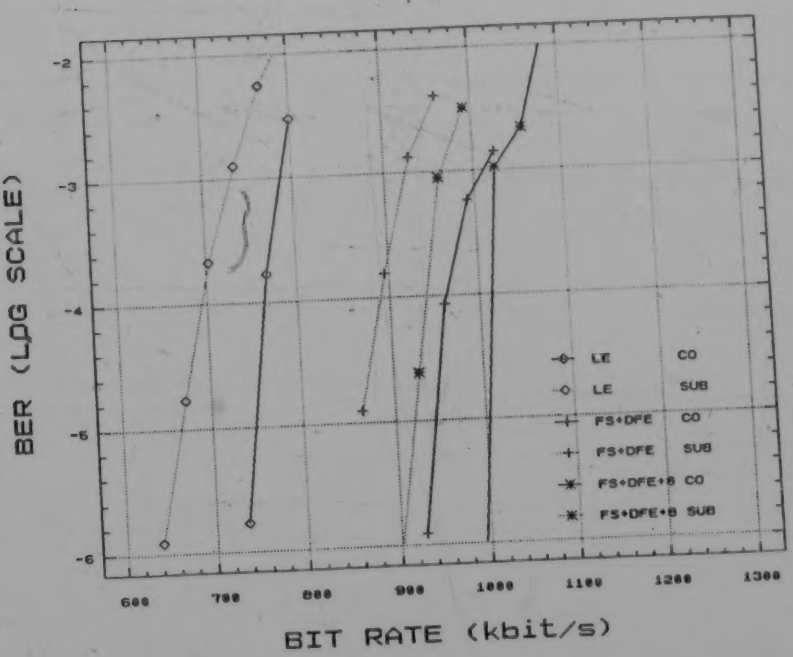


Figure 5.6 BER versus bit rate curves for the uncoded 16 QAM and coded 32 QAM passband transceivers transmitting on the loop in Figure 5.2a .

However, comparing the transmission rates of the best baseband transceivers to the best passband transceivers, we now find the baseband transceivers are performing better. The transmission rate on the central office side for the baseband transceiver, without trellis encoding and with the composite FSLE and DFE, is approximately 975 kb/s. The same structure in the passband has a performance of 940 kb/s. This also holds for the trellis coded transceivers where 1080 kb/s is achieved in the baseband compared to approximately 1000 kb/s in the passband.

Once again, even for this worst case line, a transmission rate of 800 kb/s seems feasible in the CSA. However, for this last loop, the performance of the composite FSLE and DFE appears to just satisfy the requirements of full-duplex transmission rate at a BER of 10^{-7}. With this small performance margin, it is likely that there exists a loop for which, at a BER of 10^{-7}, the maximum bit rate is less than 800 kb/s. The eight state trellis coded transceiver, on the other hand, has a considerable performance margin at this BER. Consequently, it is reasonable to suggest that trellis coded baseband transceivers may provide the added performance necessary to enable the successful deployment of the high-bit-rate digital subscriber loop access over the twisted pair CSA.

Chapter 6

Conclusions

This thesis has analyzed different transceiver structures for the proposed high-bit-rate (800 kb/s) digital subscriber loop interface which would operate on loops in the CSA. The investigation has included examining the possibilities of increasing the transmission rate on these metallic loops through the use of trellis coded modulation and improved equalization algorithms. The actual analysis was accomplished through an extensive simulation study which included models of the subscriber loops, the various transceivers and a number of noise sources. These noise sources included echo, coloured Gaussian noise and near-end crosstalk, which together, duplicate the noise environment of an actual digital subscriber loop system. The performances of different transceivers were compared by evaluating their bit-error-rates at different transmission rates.

In general, we compared the achievable bit rates of transceivers using 8 PAM, 16 QAM and 32 QAM signalling combined with optimal four and eight state trellis codes to the achievable bit rates of transceivers using uncoded 4 PAM, 8 AMPM and 16 QAM, respectively. These comparisons were also done with a variety of different equalizer structures. From our results, we have found that by using both a T/2 FSLE and a DFE together, system performance can be dramatically improved over using either equalizer separately. In addition to this, using an ISI canceller to perform the equalization does not give any improvement over the former equalizer structure. We also found, that the uncoded 8 AMPM and coded 16 QAM transceivers had the worst performances compared to the other uncoded and coded transceivers, respectively. The best

performance curves were obtained from the baseband transceivers while the uncoded 16 QAM and coded 32 QAM transceivers were a close second. Since the passband transceivers are considerably more complicated, the baseband transceivers are clearly the transceivers which should be used to provide digital transmission over subscriber loops. However, even the baseband transceivers which use the more sophisticated equalization but no coding cannot provide the high-bit-rate access (800 kb/s) with adequate margin to ensure reliable operation. Fortunately, from our results, we have found that when trellis encoding (eight state) is introduced into these transceivers using the T/2 FSLE and DFE, they achieve the gain in performance which is necessary to ensure the ubiquitous deployment of the high-bit-rate access in the CSA.

Some suggestions for future work:

- -Find and include an appropriate model for impulse noise.
- -Investigate the cyclostationarity of NEXT so as to determine whether a reduction in the noise power of this disturbance is possible by sampling at appropriate instants in time.
- -Double check the influence of FEXT, especially when operating at higher bit rates than 800 kb/s.
- -See if the LMS algorithm described in [37] will eliminate the need to guess at an optimum step size μ.
- -Introduce a noise whitening filter and see if it will improve the performance of the Viterbi decoder and PDFD.
- -Use a parallel decision feedback decoder with additional states to eliminate some of the ISI on the signal.

-Examine the performance of baseband transceivers which transmit more than 2 bits/sec/Hz so as to investigate the possibility of achieving primary rate access (1.544 Mb/s) on most of the CSA.

REFERENCES

[1] N.S. Lin and C.P. Tzeng, "Full-duplex data over local loops", IEEE Comm. Mag., Vol.26, No.2, pp 31-42, Feb. 1988.

[2] C.L. Wong and R. Wood, "Implementation of ISDN", Telesis, Vol. 3, pp 5-13, 1986.

[3] K. Feher, Advanced Digital Communications, Prentice-Hall Inc., 1987.

[4] S.N. Pandhi, "The universal data connection", IEEE Spectrum, pp 31-37, July 1987.

[5] J.T. Aslanis and J.M. Cioffi, "Capacity and cutoff rates of the digital subscriber loop with near end crosstalk noise", ICC'89, pp 35.1.1-35.1.5, June 1989.

[6] J.A. Guinea et al., "Digital transmission in the subscriber loop", IEEE Circuits and Devices Magazine, pp 14-28, Sept. 1986.

[7] "Integrated Services Digital Network-Basic Access Interface for Use on Metallic Loops for Applications on the Network Side of the NT-(Layer 1 Specification)", ANSI Standard T1.601-1988.

[8] J.W. Lechleider, "The feasibility and advisability of a DSL operating at substantially higher information rates than that of the Basic Access Interface", ECSA Document T1E1.4/88-038.

[9] V. Joshi and D.D. Falconer, "Channel capacity bounds for the subscriber loop", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp 202-204, June 4-5, 1987.

[10] R.G. Hunt et al., "The potential for high-rate digital subscriber loops" International Conference on Communications '89, pp 17.1.1-17.1.6, June 1989.

[11] S. Yamano et al., "Design philosophy and performance for ISDN basic access digital subscriber loops", Proc. of ICC, 1987.

[12] 1981 Bell Canada Subscriber Loop Survey, May 1983.

[13] J.W. Lechleider, "Loop Transmission Aspects of ISDN Basic Access", IEEE J. on Selected Areas in Comm., vol. sac-4, no.8, pp 1294-1301, Nov., 1986.

[14] R. Ziemer and W. Tranter, Principles of Communications, Houghton Mifflin Company, 2nd edition, 1985.

[15] S.U.H. Qureshi, "Adaptive Equalization", Proceedings of the IEEE, Vol.73, No. 9, pp 1349-1387, Sept. 1985.

[16] A. Gersho and T.L. Lim, "Adaptive cancellation of intersymbol interference for data transmission", Bell Systems Tech. J., vol. 60, pp. 1997-2021, Nov. 1981.

[17] J.G. Proakis, Digital Communications, McGraw Hill, 1983.

[18] E.A. Lee and D.G. Messerschmitt, Digital Communication, Kluwer Academic Publishers, 1988.

[19] I. Korn, Digital Communications, Van Nostrand Reinhold Company, 1985.

[20] J.R. Treichler et al., Theory and Design of Adaptive Filters, John Wiley and Sons, 1987.

[21] E. Arnon et al., "Transmission system for ISDN loops", Telesis, vol.3, pp. 35-45, 1986.

[22] K. Szechenyi and K. Bohm, "Impulsive noise limited transmission performance of ISDN subscriber loops", Proc. of the ISSLS'88, pp 29-34.

[23] A. Fung et al., "A facility for near-end crosstalk measurements on ISDN subscriber loops", Globecom '89, Nov. 1989.

[24] J.C. Campbell et al., "The cyclostationary nature of crosstalk interference from digital signals in multipair cable- Part I: fundamentals", IEEE Trans. on Comm., Vol. Com-31, pp. 638-649, May 1983.

[25] AT&T Bell Laboratories, Transmission Systems for Communications, Bell Telephone Laboratories Inc., 1982.

[26] G. Ungerboeck, "Channel coding with multilevel/phase signals", IEEE Trans. on Inf. Theory, Vol. IT-28, pp. 55-67, Jan. 1982.

[27] K. Wesolowski, "Efficient digital receiver structure for trellis-coded signals transmitted through channels with intersymbol interference", Electronics Letters, Vol. 23, pp. 1265-1267, Nov. 19, 1987.

[28] P.R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise", IEEE Trans. on Comm., Vol. 37, pp. 669-676, July 1989.

[29] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels", IEEE J. on Sel. Areas in Comm., Vol. 7, pp. 989-995, Aug. 1989.

[30] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets. Part II: State of the Art", IEEE Comm. Mag., Vol. 25, pp. 12-21, Feb. 1987.

[31] L.F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space-Part I: 180°", IEEE J. on Sel. Areas in Comm., Vol. SAC-2, pp. 659-671, Sept. 1984.

[32] L.F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space-Part II: Nonlinear Codes", IEEE J. on Sel. Areas in Comm., Vol. SAC-2, pp. 672-686, Sept. 1984.

[33] D.D. Falconer, Private Communication, 1987.

[34] T.D. Bilodeau, "Programs accompanying the thesis 'Trellis coded modulation for high bit rate digital subscriber loops.'", ATRC Internal Technical Report - TR-90-21, Jan. 1990.

[35] W.T. Vetterling et al., Numerical Recipes, Cambridge University Press, 1985.

[36] M.C. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems", IEEE J. on Sel. Areas in Comm., Vol. SAC-2, pp 153-170, Jan. 1984.

[37] S. Chung and P. McLeod, "Stochastic LMS with self adaptive forgetting factor", IEEE Pacific Rim Conference on Comm., Computers and Signal Processing, pp 447-450, June 1989.

[38] J.M. Smith, Mathematical modeling and digital simulation for engineers and scientists, John Wiley and Sons Inc., 2nd edition, 1987.

Appendix A

Derivation of Critical Frequencies

The input impedance of a bridged tap as seen from a main loop is given by

$$Z_{in} = Z_0 \left[\frac{Z_L + Z_0 \tanh(\gamma \ell)}{Z_0 + Z_L \tanh(\gamma \ell)} \right] \qquad (A.1)$$

where Z_L is the termination impedance on the end of the tap, ℓ is the length of the tap and Z_0 and γ are the characteristic impedance and propagation constant of the bridged tap, respectively. Z_0 and γ can be calculated using Eqns.(4.1) and (4.2). If Z_L approaches infinity, as is the case for an open-circuited bridged tap, Eqn.(A.1) becomes

$$Zin = \frac{Z_0}{\tanh(\gamma \ell)} \qquad (A.2)$$

It is now important to note that the hyperbolic tangent function will approach infinity whenever the imaginary part of its argument approaches $\pi/2$ or π multiples of it. From Eqn.(A.2), this results in Z_{in} approaching zero which means that the bridged tap will appear to be a short circuit as seen from the main loop. The imaginary part of γ is the phase constant β which is given by

$$\beta = [0.5 \{ (R^2+(2\pi fL)^2)\ (G^2+(2\pi fC)^2) \}^{0.5} - RG + (2\pi f)^2 LC]^{0.5} \qquad (A.3)$$

Consequently, Z_{in} will equal zero when

$$\beta_n \ell = \pi/2 + n\pi \qquad n=0,1,2,\ldots \qquad (A.4)$$

We should now note that the velocity of propagation on the bridged tap is given by

$$u = \lambda f = \frac{2\pi f}{\beta} \qquad (A.5)$$

where f is the frequency and λ is the wavelength. From this, we see that the bridged tap will appear to be a short circuit to the main cable pair when the length ℓ of the bridged tap is given by

$$2\pi\ell/\lambda_n = \pi/2+n\pi$$

$$\ell = \lambda_n/4 + n\lambda_n/2 \qquad n=0,1,2,... \tag{A.6}$$

That is, the bridged tap will be a short circuit whenever its length is equal to a quarter wavelength or a half wavelength multiple of a quarter wavelength at some frequency. The critical frequencies associated with these wavelengths can be found by using Eqns.(A.5) and (A.6) to give, after some manipulation

$$f_n = \frac{u\ (1+2n)}{4\ell} \qquad n=0,1,2,... \tag{A.7}$$

where for any one bridge tap length ℓ, there will be an infinite number of critical frequencies.

This can be demonstrated by an example. Figure A.1 shows the magnitude response of a 2100 m 26 AWG cable which has two 450 m 24 AWG bridged taps. We can use Eqns.(A.3), (A.5) and (A.7) to predict where the critical or dip frequencies will be. The first step is to use Eqns.(A.3) and (A.5) to find the velocity of propagation along the bridged taps. The values we require for R, L, G and C can be found in Appendix B. Table B.2, for the 24 AWG cable at 21° Celsius, shows us that C is constant at 0.0516 μF/km but R, L and G are functions of frequency. Since we are looking for the critical frequencies, we do not know which values of these primary parameters to use. We can handle this problem by choosing any initial values for R, L and G and then re-iterating the calculation with new values which correspond to frequencies which are closer to the critical frequencies. For the first

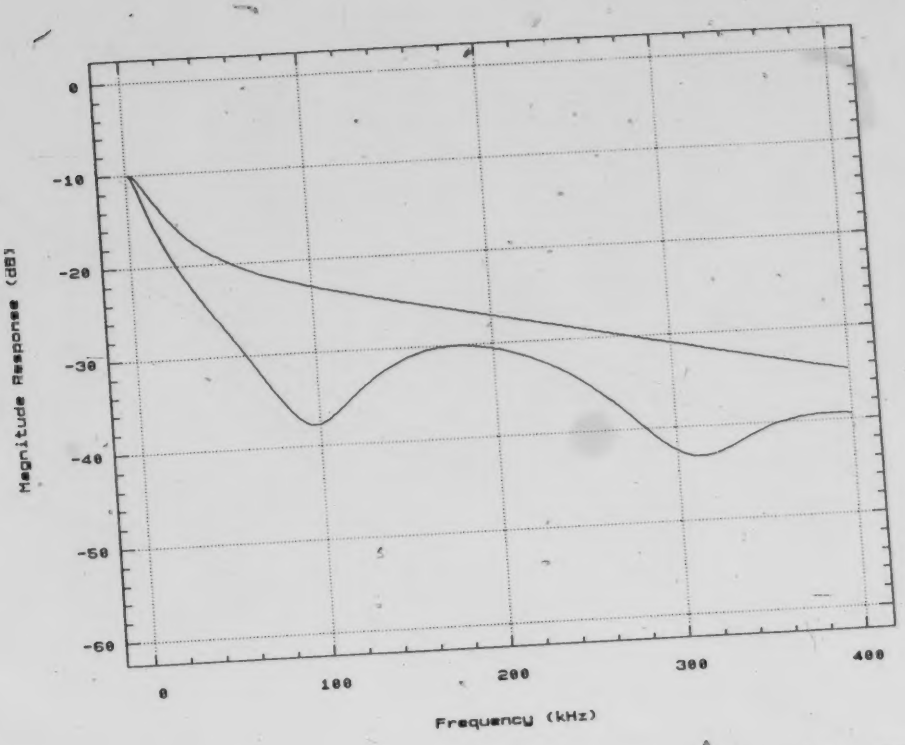


Figure A.1 Magnitude response of a 2100m 26 AWG cable with two 450m 24 AWG bridged taps

iteration, we can choose a frequency of 200 kHz which corresponds to R=229.3 Ω/km, L=0.5646 mH/km and G=7.177 μS/km. Substituting these values for R, L, G and C into Eqn.(A.3) and then the value for β into Eqn.(A.5), results in $u=1.83*10^5$ km/s or $1.83*10^8$ m/s. Finally, substituting this value, along with ℓ=450 m, into Eqn.(A.7) results in the critical frequencies f_0=102 kHz, f_1=305 kHz, f_2=508 kHz, etc.. We may now re-iterate these calculations but now using these approximate values for the critical frequencies as a guide in choosing new values for R, L and G.

For example, at the frequency of 100 kHz, R=191.64 Ω/km, L=0.5808 mH/km and G=3.927 μS/km. Using these values in the above calculation, results in f_0=98 kHz Similarly, at 300 kHz, the values of R=268.15 Ω/km, L=0.5521 mH/km and G=10.213 μS/km give the critical frequency f_1=310 kHz. Finally, the same procedure, at 500 kHz, leads to f_2=527 kHz. By comparing these values to the previously calculated values for the critical frequencies, we see that another iteration is not necessary.

We may now compare these values for the critical frequencies to the locations of the frequency dips in Figure A.1. We see that the first dip, in the figure, is at approximately 96 kHz while the second dip is at about 315 kHz. These values are very close to our calculated frequencies of 98 kHz and 310 kHz, respectively. From this, we can be quite confident that if the frequency axis was extended to larger values of frequency, we would see a third dip at approximately 527 kHz and additional dips at frequencies which satisfy Eqn.(A.7).

Appendix B

Tables of Primary Constants

These are the tables of primary constants for the 19, 22, 24 and 26 AWG PIC cables at 21° Celsius which were used in the simulations. Additional tables, for different temperatures and insulation type, are given in [7].

TABLE B.1

Primary Constants (1 Hz to 5 MHz)

of 19 AWG PIC at 21 Degrees Celsius

FREQ (Hz)	R Ω/km	L mH/km	G μS/km	C μF/km
1.	54.02	0.6127	0.000	0.0516
5.	54.02	0.6127	0.001	0.0516
10.	54.02	0.6127	0.001	0.0516
15.	54.02	0.6127	0.002	0.0516
20.	54.02	0.6127	0.002	0.0516
30.	54.02	0.6127	0.004	0.0516
50.	54.02	0.6127	0.006	0.0516
70.	54.02	0.6127	0.008	0.0516
100.	54.02	0.6127	0.012	0.0516
150.	54.03	0.6127	0.017	0.0516
200.	54.03	0.6127	0.022	0.0516
300.	54.03	0.6126	0.033	0.0516
500.	54.03	0.6125	0.054	0.0516
700.	54.04	0.6124	0.075	0.0516
1000.	54.05	0.6122	0.104	0.0516
1500.	54.08	0.6120	0.153	0.0516
2000.	54.10	0.6117	0.202	0.0516
3000.	54.16	0.6112	0.298	0.0516
5000.	54.32	0.6102	0.485	0.0516
7000.	54.52	0.6091	0.668	0.0516
10000.	54.90	0.6075	0.939	0.0516
15000.	55.76	0.6047	1.383	0.0516
20000.	56.86	0.6019	1.819	0.0516
30000.	59.67	0.5922	2.678	0.0516
50000.	66.87	0.5796	4.360	0.0516
70000.	74.83	0.5699	6.010	0.0516
100000.	86.24	0.5575	8.446	0.0516
150000.	103.46	0.5410	12.435	0.0516
200000.	118.00	0.5286	16.362	0.0516
300000.	142.41	0.5131	24.090	0.0516
500000.	181.10	0.4969	39.217	0.0516
700000.	212.56	0.4879	54.061	0.0516
1000000.	252.21	0.4796	75.973	0.0516
1500000.	306.77	0.4717	111.854	0.0516
2000000.	352.76	0.4668	147.178	0.0516
3000000.	429.93	0.4609	216.688	0.0516
5000000.	552.29	0.4547	352.759	0.0516

TABLE B.2

Primary Constants (1 Hz to 5 MHz)

of 22 AWG PIC at 21 Degrees Celsius

FREQ (Hz)	R Ω/km	L mH/km	G μS/km	C μF/km
1.	108.29	0.6127	0.000	0.0516
5.	108.29	0.6127	0.001	0.0516
10.	108.29	0.6127	0.001	0.0516
15.	108.29	0.6127	0.001	0.0516
20.	108.29	0.6127	0.001	0.0516
30.	108.29	0.6127	0.002	0.0516
50.	108.29	0.6127	0.003	0.0516
70.	108.29	0.6127	0.004	0.0516
100.	108.29	0.6127	0.006	0.0516
150.	108.29	0.6127	0.008	0.0516
200.	108.29	0.6127	0.011	0.0516
300.	108.29	0.6127	0.015	0.0516
500.	108.30	0.6125	0.025	0.0516
700.	108.30	0.6125	0.034	0.0516
1000.	108.31	0.6124	0.047	0.0516
1500.	108.33	0.6122	0.068	0.0516
2000.	108.35	0.6121	0.090	0.0516
3000.	108.39	0.6117	0.131	0.0516
5000.	108.50	0.6110	0.212	0.0516
7000.	108.63	0.6102	0.290	0.0516
10000.	108.88	0.6092	0.405	0.0516
15000.	109.40	0.6073	0.593	0.0516
20000.	110.05	0.6055	0.775	0.0516
30000.	111.76	0.6010	1.133	0.0516
50000.	116.59	0.5897	1.829	0.0516
70000.	122.85	0.5823	2.505	0.0516
100000.	133.94	0.5740	3.498	0.0516
150000.	153.83	0.5627	5.113	0.0516
200000.	172.71	0.5529	6.693	0.0516
300000.	207.16	0.5370	9.783	0.0516
500000.	261.95	0.5163	15.780	0.0516
700000.	306.49	0.5047	21.621	0.0516
1000000.	362.63	0.4940	30.191	0.0516
1500000.	439.87	0.4836	44.126	0.0516
2000000.	505.00	0.4773	57.761	0.0516
3000000.	614.24	0.4696	84.423	0.0516
5000000.	787.47	0.4616	136.178	0.0516

TABLE B.3

Primary Constants (1 Hz to 5 MHz)

of 24 AWG PIC at 21 Degrees Celsius

FREQ (Hz)	R Ω/km	L mH/km	G μS/km	C μF/km
1.	172.24	0.6127	0.000	0.0516
5.	172.24	0.6127	0.001	0.0516
10.	172.24	0.6127	0.001	0.0516
15.	172.24	0.6127	0.002	0.0516
20.	172.24	0.6127	0.002	0.0516
30.	172.24	0.6127	0.003	0.0516
50.	172.24	0.6127	0.005	0.0516
70.	172.24	0.6127	0.007	0.0516
100.	172.24	0.6127	0.010	0.0516
150.	172.24	0.6127	0.014	0.0516
200.	172.24	0.6127	0.017	0.0516
300.	172.24	0.6127	0.025	0.0516
500.	172.25	0.6126	0.039	0.0516
700.	172.26	0.6125	0.052	0.0516
1000.	172.26	0.6125	0.071	0.0516
1500.	172.28	0.6123	0.102	0.0516
2000.	172.29	0.6122	0.130	0.0516
3000.	172.33	0.6119	0.186	0.0516
5000.	172.42	0.6114	0.290	0.0516
7000.	172.53	0.6107	0.388	0.0516
10000.	172.72	0.6099	0.530	0.0516
15000.	173.10	0.6085	0.754	0.0516
20000.	173.58	0.6071	0.968	0.0516
30000.	174.79	0.6042	1.378	0.0516
50000.	178.22	0.5951	2.149	0.0516
70000.	182.86	0.5881	2.879	0.0516
100000.	191.64	0.5808	3.927	0.0516
150000.	209.54	0.5719	5.588	0.0516
200000.	229.30	0.5646	7.177	0.0516
300000.	268.15	0.5521	10.213	0.0516
500000.	336.59	0.5325	15.928	0.0516
700000.	392.76	0.5188	21.345	0.0516
1000000.	463.57	0.5062	29.111	0.0516
1500000.	561.00	0.4938	41.424	0.0516
2000000.	643.14	0.4862	53.204	0.0516
3000000.	780.92	0.4770	75.708	0.0516
5000000.	999.40	0.4675	118.074	0.0516

TABLE B.4

Primary Constants (1 Hz to 5 MHz)

of 26 AWG PIC at 21 Degrees Celsius

FREQ (Hz)	R Ω/km	L mH/km	G μS/km	C μF/km
1.	273.87	0.6127	0.000	0.0516
5.	273.87	0.6127	0.001	0.0516
10.	273.87	0.6127	0.001	0.0516
15.	273.88	0.6127	0.002	0.0516
20.	273.88	0.6127	0.002	0.0516
30.	273.88	0.6127	0.003	0.0516
50.	273.88	0.6127	0.005	0.0516
70.	273.88	0.6127	0.007	0.0516
100.	273.88	0.6127	0.010	0.0516
150.	273.88	0.6127	0.014	0.0516
200.	273.88	0.6127	0.017	0.0516
300.	273.88	0.6127	0.025	0.0516
500.	273.88	0.6126	0.039	0.0516
700.	273.89	0.6126	0.052	0.0516
1000.	273.89	0.6125	0.071	0.0516
1500.	273.91	0.6124	0.102	0.0516
2000.	273.92	0.6123	0.130	0.0516
3000.	273.95	0.6121	0.186	0.0516
5000.	274.03	0.6116	0.290	0.0516
7000.	274.12	0.6112	0.388	0.0516
10000.	274.27	0.6105	0.530	0.0516
15000.	274.57	0.6094	0.754	0.0516
20000.	274.93	0.6083	0.968	0.0516
30000.	275.81	0.6060	1.378	0.0516
50000.	278.26	0.6002	2.149	0.0516
70000.	281.54	0.5932	2.879	0.0516
100000.	287.94	0.5861	3.927	0.0516
150000.	301.86	0.5782	5.588	0.0516
200000.	318.79	0.5724	7.177	0.0516
300000.	358.02	0.5631	10.213	0.0516
500000.	434.72	0.5478	15.928	0.0516
700000.	505.14	0.5352	21.345	0.0516
1000000.	594.43	0.5208	29.111	0.0516
1500000.	717.30	0.5062	41.424	0.0516
2000000.	820.87	0.4972	53.204	0.0516
3000000.	994.62	0.4861	75.708	0.0516
5000000.	1270.13	0.4746	118.074	0.0516

Appendix C

Derivation of an Equivalent Series ABCD Matrix for a Bridged Tap

We begin by considering a mixed gauge bridged tap which has an in parallel composite ABCD matrix given by

$$\begin{bmatrix} A_{comp} & B_{comp} \\ C_{comp} & D_{comp} \end{bmatrix} \tag{C.1}$$

This matrix leads to the equations

$$V_3 = A_{comp}V_4 + B_{comp}I_4 \tag{C.2}$$

$$I_3 = C_{comp}V_4 + D_{comp}I_4 \tag{C.3}$$

where $V_4/I_4 = Z_L$ as shown in Figure C.1 . By examining this figure, we will also observe that the equivalent admittance of the bridged tap, as seen from the main line, can be expressed as

$$Y_{eq} = \frac{I_3}{V_3} = \frac{C_{comp} \cdot Z_L + D_{comp}}{A_{comp} \cdot Z_L + B_{comp}} \tag{C.4}$$

Now, for an open circuited bridged tap, the value of Z_L will equal infinity. By substituting this into Eqn.(C.4) we get the result.

$$Y_{eq} = C_{comp}/A_{comp} \tag{C.5}$$

The next step in the derivation will be to redraw Figure C.1, except now we will represent the bridged tap as an in series two port network which consists of the equivalent shunt impedance Y_{eq}. This is shown in Figure C.2 . By noting that $V_2 = V_3 = V_5$ and by taking the sum of currents at node 3, we get the following two equations

$$V_2 = V_5 \tag{C.6}$$

$$I_2 = Y_{eq}V_5 + I_5 \tag{C.7}$$

If we compare Eqns.(C.6) and (C.7) to the general description of this two-port network given by

$$V_2 = A_{sbr}V_5 + B_{sbr}I_5 \tag{C.8}$$

$$I_2 = C_{sbr}V_5 + D_{sbr}I_5 \tag{C.9}$$

respectively, we find that the equivalent series ABCD matrix for the bridged tap is equal to

$$\begin{bmatrix} A_{sbr} & B_{sbr} \\ C_{sbr} & D_{sbr} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ Y_{eq} & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ C_{comp}/A_{comp} & 1 \end{bmatrix} \tag{C.10}$$

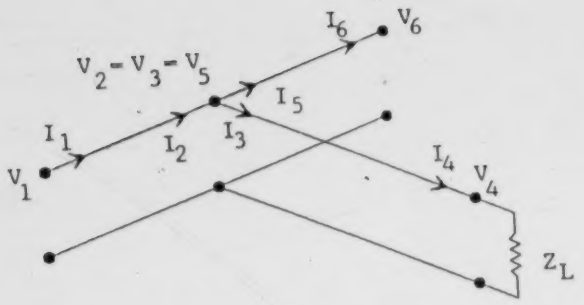


Figure C.1 Main loop with a bridged tap

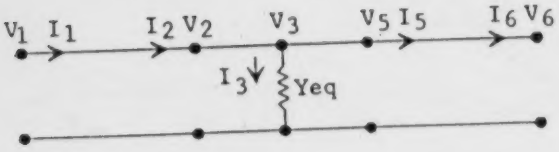


Figure C.2 Bridged tap as a series two-port network

Appendix D

Composite ABCD Matrices for the Two Directions of Transmission

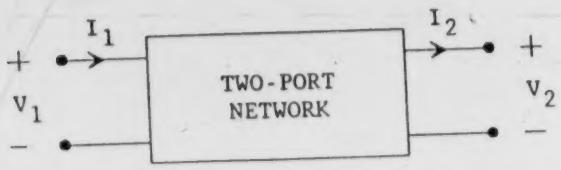


Figure D.1 Two-port network for the direction of transmission from the line termination to the network termination

Figure D.1 shows a two-port network which we may assume illustrates the transmission of a signal from the line termination to the network termination of some subscriber loop. The composite ABCD matrix of this loop for this direction of transmission could be generated by the program DTXLINE.FOR [34] and would give the relationships

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A_{comp} & B_{comp} \\ C_{comp} & D_{comp} \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \qquad (D.1)$$

Essentially, this ABCD matrix is used to express the values of the input voltage and current on the LT side in terms of the output voltage and current on the NT side. Hence, it follows that the ABCD matrix for the opposite direction of transmission will express the voltage and current, V_2 and I_2, in terms of the voltage and current V_1 and I_1. After some manipulation of Eqn.(D.1), we can get the following relationships.

$$\begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \frac{1}{A_{comp}D_{comp} - B_{comp}C_{comp}} \begin{bmatrix} D_{comp} & -B_{comp} \\ -C_{comp} & A_{comp} \end{bmatrix} \begin{bmatrix} V_1 \\ I_1 \end{bmatrix} \qquad (D.2)$$

At this point, Eqn.(D.2) can be simplified by showing that $A_{comp}D_{comp}-B_{comp}C_{comp}$, which is the determinant of the composite ABCD matrix, is always equal to unity.

We may begin by calculating the determinant of an ABCD matrix for a uniform section of cable. Such a matrix is given by

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \cosh(\gamma L) & Z_0\sinh(\gamma L) \\ \sinh(\gamma L)/Z_0 & \cosh(\gamma L) \end{bmatrix} \tag{D.3}$$

where Z_0 is the characteristic impedance, γ is the propagation constant and L is the length of the cable section. The determinant of this ABCD matrix is given by

$$\begin{aligned} \det \begin{bmatrix} A & B \\ C & D \end{bmatrix} &= AD - BC \\ &= \cosh^2(\gamma L) - \sinh^2(\gamma L) \\ &= 1 \end{aligned} \tag{D.4}$$

Hence, the determinant of an ABCD matrix for a uniform section of cable is always equal to unity. Next, we may calculate the determinant of an equivalent series ABCD matrix for a bridged tap. This matrix is derived in Appendix C and is given by

$$\begin{bmatrix} A_{sbr} & B_{sbr} \\ C_{sbr} & D_{sbr} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C_{compbr}/A_{compbr} & 1 \end{bmatrix} \tag{D.5}$$

where C_{compbr} and A_{compbr} are the A and C elements of the parallel composite ABCD matrix for a bridged tap. The determinant of this matrix is given by

$$\det \begin{bmatrix} A_{sbr} & B_{sbr} \\ C_{sbr} & D_{sbr} \end{bmatrix} = 1 - 0 = 1 \tag{D.6}$$

From Eqns.(D.4) and (D.6), we see that each of the individual sections which can be used to build a subscriber loop, have ABCD matrices with

determinants that are equal to unity. Now, by knowing the property given by

$$\det(MN) = \det(M)\cdot\det(N) \tag{D.7}$$

where M and N can be any square matrices, we can clearly see that all composite ABCD matrices will have a determinant equal to one. This is regardless as to the number or type of sections used to construct the loop. Hence, Eqn.(D.2) becomes

$$\begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \begin{bmatrix} D_{comp} & -B_{comp} \\ -C_{comp} & A_{comp} \end{bmatrix} \begin{bmatrix} V_1 \\ I_1 \end{bmatrix} \tag{D.8}$$

Finally, by examining Figure D.2 which shows the two port network for the transmission from the NT to the LT, we see that the currents I_1 and I_2 should flow in the direction of transmission. This simply implies replacing I_1 with $-I_1$ and I_2 with $-I_2$ in Eqn.(D.8) which, after some manipulation, results in

$$\begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \begin{bmatrix} D_{comp} & B_{comp} \\ C_{comp} & A_{comp} \end{bmatrix} \begin{bmatrix} V_1 \\ I_1 \end{bmatrix} \tag{D.9}$$

where the assumed current directions are as shown in Figure D.2. From this derivation, we have found that the determinant of an ABCD matrix is equal to unity and that the ABCD matrices for the two directions of transmission can be obtained from one another by simply interchanging their A_{comp} and D_{comp} terms.

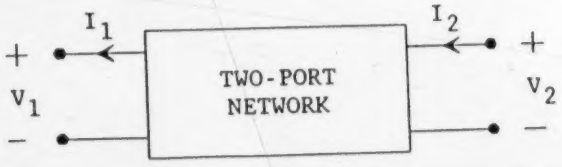


Figure D.2 Two-port network for the direction of transmission from the network termination to the line termination

Appendix E

Derivation of the Transfer Functions Involving the Subscriber Loop and the Hybrids

E.1 Transfer Function of the Signal Path through the Subscriber Loop and Hybrids.

Figure E.1 shows two hybrid models connected on either side of a twoport network which could represent any subscriber loop. In this subsection we will first obtain the transfer function of the signal's path through the two hybrids and the subscriber loop in the direction from the central office to the subscriber. We will then derive the same transfer function but in the opposite direction of transmission using the twoport network found in Appendix D and shown directly below the first network. By using the superposition theorem, we can set $V_{SUB}=0$ and only consider the source V_{CO}. We may now begin the derivation for the first direction by stating that the desired transfer function is the ratio

$$\frac{V_{RSUB}}{V_{CO}} = \frac{V_1}{V_{CO}} \cdot \frac{V_2}{V_1} \cdot \frac{V_{RSUB}}{V_2} \qquad (E.1)$$

where all of these voltages are shown in the figure. The voltage V_1 at the line termination interface is given by

$$V_1 = G_{CO} V_{CO} \frac{Z_{CO}}{R_L + Z_{CO}} \qquad (E.2)$$

where G_{CO} is the gain of the central office's (CO) line driver, R_L is the load resistance and Z_{CO} is the equivalent impedance of the loop as seen from the CO side. This can be re-written to give the transfer function

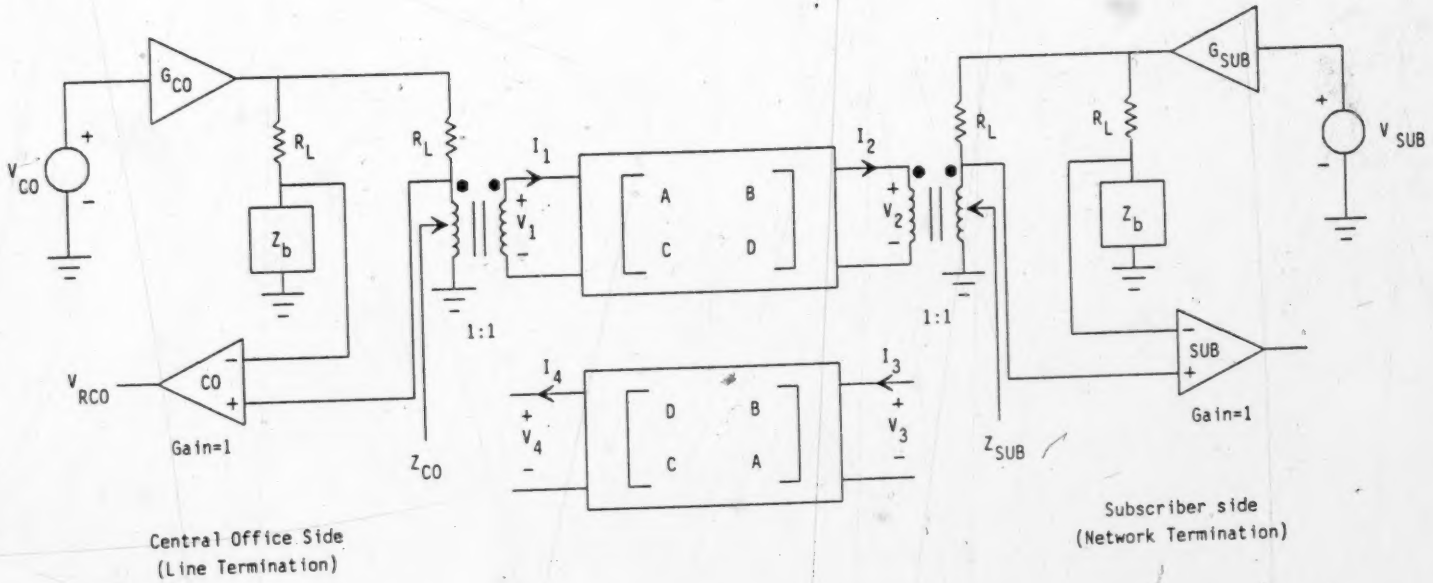


Figure E.1 A model for a subscriber loop and hybrids

$$\frac{V_1}{V_{CO}} = G_{CO}\,\frac{Z_{CO}}{R_L+Z_{CO}} \tag{E.3}$$

Now, by assuming the output impedance of the subscriber's (SUB) line driver is very small (ideally zero), the line driver's output will be a virtual ground. This results in the two relationships

$$V_2 = R_L\, I_2 \tag{E.4}$$

and

$$V_{RSUB}=V_2 \;\rightarrow\; V_{RSUB}/V_2 = 1 \tag{E.5}$$

From Eqn.(E.4) and the two fundamental equations

$$V_1 = A\,V_2 + B\,I_2 \tag{E.6}$$

$$I_1 = C\,V_2 + D\,I_2 \tag{E.7}$$

we can obtain an expression for the equivalent impedance Z_{CO} which is given by

$$Z_{CO} = \frac{V_1}{I_1} = \frac{AR_L+B}{CR_L+D} \tag{E.8}$$

Finally, the voltage V_2 at the network termination interface can be found in terms of V_1 by manipulating Eqns.(E.6) and (E.7) to get

$$V_2 = \frac{DV_1-BI_1}{AD-BC} = V_1(D-B/Z_{CO}) \tag{E.9}$$

since AD-BC = 1 (determinant of an ABCD matrix) and $Z_{CO}=V_1/I_1$. By placing the transfer functions given by Eqns.(E.3), (E.5) and (E.9) into Eqn.(E.1), we get the result that

$$\frac{V_{RSUB}}{V_{CO}} = \left[\, G_{CO}\,\frac{Z_{CO}}{R_L+Z_{CO}} \,\right]\left[\, D - \frac{B}{Z_{CO}} \,\right]$$

$$= \frac{G_{CO}(DZ_{CO}-B)}{R_L+Z_{CO}} \tag{E.10}$$

If we now substitute the expression for Z_{CO} given by Eqn.(E.8) into Eqn.(E.10), the transfer function of the signal path from the CO to the SUB, through both the hybrids and a subscriber loop, is given by

$$\frac{RSUB}{V_{CO}} = \frac{G_{CO}\, R_L}{AR_L + B + CR_L^2 + DR_L} \qquad (E.11)$$

The signal path transfer function for a transmission from SUB side to the CO side can be found in a very similar way. However, for this direction, the lower twoport network labeled with the voltages V_3 and V_4 and the currents I_3 and I_4 will be used to represent the subscriber loop. To start this derivation we can state that the desired transfer function is given by the ratio

$$\frac{V_{RCO}}{V_{SUB}} = \frac{V_3}{V_{SUB}} \cdot \frac{V_4}{V_3} \cdot \frac{V_{RCO}}{V_4} \qquad (E.12)$$

where these voltages are also shown in Figure E.1 . The transfer function V_3/V_{SUB} can be obtained in a similar manner as V_1/V_{CO} shown previously and is thus given by

$$\frac{V_3}{V_{SUB}} = G_{SUB} \frac{Z_{SUB}}{R_L + Z_{SUB}} \qquad (E.13)$$

where G_{SUB} is the gain of the SUB's line driver, R_L is again the load resistance and Z_{SUB} is the equivalent impedance of the loop as seen from the SUB side. Since the output of the CO's line driver will be a virtual ground (as was the SUB's) we get the equations

$$V_4 = R_L\, I_4 \qquad (E.14)$$

and

$$V_{RCO}/V_4 = 1 \qquad (E.15)$$

Now, from Eqn.(E.14) and the two fundamental relationships

$$V_3 = D\,V_4 + B\,I_4 \tag{E.16}$$

$$I_3 = C\,V_4 + A\,I_4 \tag{E.17}$$

the expression for Z_{SUB} is given by

$$Z_{SUB} = \frac{V_3}{I_3} = \frac{DR_L+B}{CR_L+A} \tag{E.18}$$

The final transfer function needed, V_4/V_3, can be obtained by manipulating Eqns.(E.16) and (E.17) to give

$$\frac{V_4}{V_3} = (A-B/Z_{SUB}) \tag{E.19}$$

By placing the transfer functions given by Eqns.(E.13), (E.15) and (E.19) into Eqn.(E.12) we get the result

$$\frac{V_{RCO}}{V_{SUB}} = \left[G_{SUB}\,\frac{Z_{SUB}}{R_L+Z_{SUB}} \right] \left[A - \frac{B}{Z_{SUB}} \right]$$

$$= \frac{G_{SUB}(AZ_{SUB}-B)}{R_L+Z_{SUB}} \tag{E.20}$$

If we now substitute the expression for Z_{SUB} given by Eqn.(E.18) into Eqn.(E.20), the transfer function of the signal path from the SUB to the CO, through both the hybrids and a subscriber loop, is given by

$$\frac{V_{RCO}}{V_{SUB}} = \frac{G_{SUB}\,R_L}{AR_L + B + CR_L^2 + DR_L} \tag{E.21}$$

This expression is the same as that given by Eqn.(E.11), except for the possibility that the line drivers have different gains.

E.2 Transfer Function of the Echo Path

The transfer functions of the echo paths (transhybrid transfer functions) on both sides of the subscriber loop, can also be found by examining Figure E.1 . By using the superposition theorem, the transfer

function of the echo path on the CO side can be found by setting $V_{SUB}=0$ and finding an expression for V_{RCO}/V_{CO}. This expression can be found by considering the voltages at the inputs of the differential amplifier. The transhybrid transfer function is given by

$$\frac{V_{RCO}}{V_{CO}} = G_{CO}\left[\frac{Z_{CO}}{R_L+Z_{CO}} - \frac{Z_b}{R_L+Z_b}\right] \qquad (E.22)$$

where Z_b is the impedance of the balancing network. By substituting the expression for Z_{CO} from Eqn.(E.8) into Eqn.(E.22), the simulated transhybrid transfer function at the CO's side is given by

$$\frac{V_{RCO}}{V_{CO}} = G_{CO}\left[\frac{AR_L+B}{AR_L + B + CR_L^2 + DR_L} - \frac{Z_b}{R_L+Z_b}\right] \qquad (E.23)$$

In a similar fashion, the echo path through the hybrid at the SUB's side can be found and is given by

$$\frac{V_{RSUB}}{V_{SUB}} = G_{SUB}\left[\frac{Z_{SUB}}{R_L+Z_{SUB}} - \frac{Z_b}{R_L+Z_b}\right] \qquad (E.24)$$

Finally, by now substituting the expression for Z_{SUB} from Eqn.(E.18) into Eqn.(E.24), the simulated transhybrid transfer function at the SUB's side is given by

$$\frac{V_{RSUB}}{V_{SUB}} = G_{SUB}\left[\frac{DR_L+B}{AR_L + B + CR_L^2 + DR_L} - \frac{Z_b}{R_L+Z_b}\right] \qquad (E.25)$$

Appendix F

Transfer Functions of the Simulation Experiments

In this appendix, we will derive the overall transfer functions of the signals and noises used in the simulation experiments. These expressions will represent the signal and noise paths up to the input of the adaptive equalizer. To calculate these expressions, we will first define transfer functions for each of the individual elements that are part of an overall path. We can then multiply appropriate elements together to produce the composite transfer functions we desire.

The square-root raised cosine transmit and receive filters are given by

$$H_1(f)=\begin{cases} 1 & 0\le|f|\le(1-\alpha)/2T \\ (0.5[1-\sin\{\pi T(f-1/2T)/\alpha\}])^{0.5} & (1-\alpha)/2T\le|f|\le(1+\alpha)/2T \\ 0 & |f|>(1+\alpha)/2T \end{cases} \quad (F.1)$$

The line drivers simply provide a gain given by

$$H_{2CO}(f)=G_{CO} \quad (F.2a)$$

$$H_{2SUB}(f)=G_{SUB} \quad (F.2b)$$

The signal path transfer function through the subscriber loop and both hybrids was derived in Appendix E and is given by

$$H_3(f) = \frac{R_L}{AR_L + B + CR_L^2 + DR_L} \quad (F.3)$$

The transhybrid transfer functions for the echo were also derived in Appendix E and are given by

$$H_{4CO}(f) = \left[\frac{AR_L+B}{AR_L + B + CR_L^2 + DR_L} - \frac{Z_b}{R_L+Z_b} \right] \quad (F.4a)$$

$$H_{4SUB}(f) = \left[\frac{DR_L+B}{AR_L + B + CR_L^2 + DR_L} - \frac{Z_b}{R_L+Z_b} \right] \quad (F.4b)$$

The voltage division that the transmitted signal experiences at the line or network interface is also required. On the CO side it is given by (see Figure E.1)

$$H_{5CO}(f) = \frac{V_1}{V_{CO}} = \frac{Z_{CO}}{R_L+Z_{CO}} \qquad \text{(F.5a)}$$

where the equivalent impedance Z_{CO} is expressed as (Eqn.(E.8))

$$Z_{CO} = \frac{AR_L+B}{CR_L+D} \qquad \text{(F.5b)}$$

Similarly, the voltage division on the SUB side is given by

$$H_{5SUB}(f) = \frac{V_3}{V_{SUB}} = \frac{Z_{SUB}}{R_L+Z_{SUB}} \qquad \text{(F.5c)}$$

where the equivalent impedance Z_{SUB} is expressed as (Eqn.(E.18))

$$Z_{SUB} = \frac{DR_L+B}{CR_L+A} \qquad \text{(F.5d)}$$

In addition to the loss caused by the transhybrid transfer function, the echo will be further suppressed by an echo canceller. The function of an echo canceller was modeled as a direct reduction of the echo level and can be expressed as

$$H_6(f) = E \text{ (constant attenuation)} \qquad \text{(F.6)}$$

The NEXT voltage transfer function is given by

$$H_7(f) = (K_{NEXT}\, f^{1.5})^{0.5} \qquad \text{(F.7)}$$

In all of the baseband transceivers, a multiple response fixed pre-equalizer was used. Its transfer function is given by

$$H_8(f) = 1-z^{-1}\Big|_{z=e^{j2\pi fT}} = 1-\cos(2\pi fT)+j\sin(2\pi fT) \qquad \text{(F.8)}$$

Now that we have the transfer functions for all the individual elements in the system, we may calculate the composite transfer

functions of the signals and noises simulated in the experiment. We will begin by showing the transfer functions used in the baseband transceivers. The transfer functions of the received signal paths on both the CO and the SUB sides are given by

$$H_{sCO}(f) = H_1^2(f)*H_{2SUB}(f)*H_3(f)*H_8(f) \quad (F.9)$$

$$H_{sSUB}(f) = H_1^2(f)*H_{2CO}(f)*H_3(f)*H_8(f) \quad (F.10)$$

respectively.

One of the noise sources modelled in the simulations was echo. The transfer functions of the echo paths on the CO and SUB sides are given by

$$H_{ECO}(f) = H_1^2(f)*H_{2CO}(f)*H_{4CO}(f)*H_6(f)*H_8(f) \quad (F.11)$$

$$H_{ESUB}(f) = H_1^2(f)*H_{2SUB}(f)*H_{4SUB}(f)*H_6(f)*H_8(f) \quad (F.12)$$

respectively.

A second noise source modelled was coloured Gaussian noise. Its transfer function on both sides of the subscriber loop is given by

$$H_{CGN}(f) = H_1(f)*H_8(f) \quad (F.13)$$

The final noise modelled was NEXT. The transfer functions of the NEXT paths on the CO and SUB sides are given by

$$H_{XCO}(f) = H_1^2(f)*H_{2CO}(f)*H_{5CO}(f)*H_7(f)*H_8(f) \quad (F.14)$$

$$H_{XSUB}(f) = H_1^2(f)*H_{2SUB}(f)*H_{5SUB}(f)*H_7(f)*H_8(f) \quad (F.15)$$

respectively.

One last pair of transfer functions used in these simulations come as a result of having to scale the transmitted powers at the line and network interfaces properly. A test signal is used which must pass through a square-root raised cosine filter and the voltage divider at the hybrid. The composite transfer functions describing these paths at the CO and SUB side are given by

$$H_{scCO}(f) = H_1(f)*H_{5CO}(f) \quad \text{(F.16)}$$

$$H_{scSUB}(f) = H_1(f)*H_{5SUB}(f) \quad \text{(F.17)}$$

These are all of the transfer functions required for the baseband transceivers. The passband transceivers have transfer functions that are very similar to these written above. In fact, the only difference between the two is that the $H_8(f)$ term which is used to show the contribution of the multiple response fixed pre-equalizer is not present in the passband transfer functions. This change simply comes from the fact that the pre-equalizer is not used in the passband transceivers. Other than this the transfer functions are all the same.